

MINI PROJECT

Data Science

Silvia Farida

silviafarida16@gmail.com

SILVIA FARIDA

★ **Hallo Everyone**



Nama Saya Silvia Farida

Saya seorang mahasiswa jurusan Informatika di Universitas Siliwangi. Dengan tekad belajar memperdalam mengenai bidang data, yang memiliki cita cita menjadi data analysts/data scientist. Setiap ahli pernah menjadi pemula. Jangan takut untuk memulai, karena langkah pertama selalu yang paling berharga. Belajar data bukan sekadar memahami angka, tapi tentang bagaimana kita bisa membaca cerita di baliknya

silviafarida16@gmail.com

Digit Classification using KNN

Project Overview

This project implements a handwritten digit classification system using the K-Nearest Neighbors (KNN) algorithm. The system can recognize digits 0-9 from handwritten images with high accuracy.

★ Key Features

- Multi-class classification for 10 digits
- Real-time prediction capabilities
- Comprehensive performance analysis
- Parameter optimization

★ Applications

- Optical Character Recognition (OCR)
- Handwriting recognition systems
- Educational demonstrations
- Template for similar projects

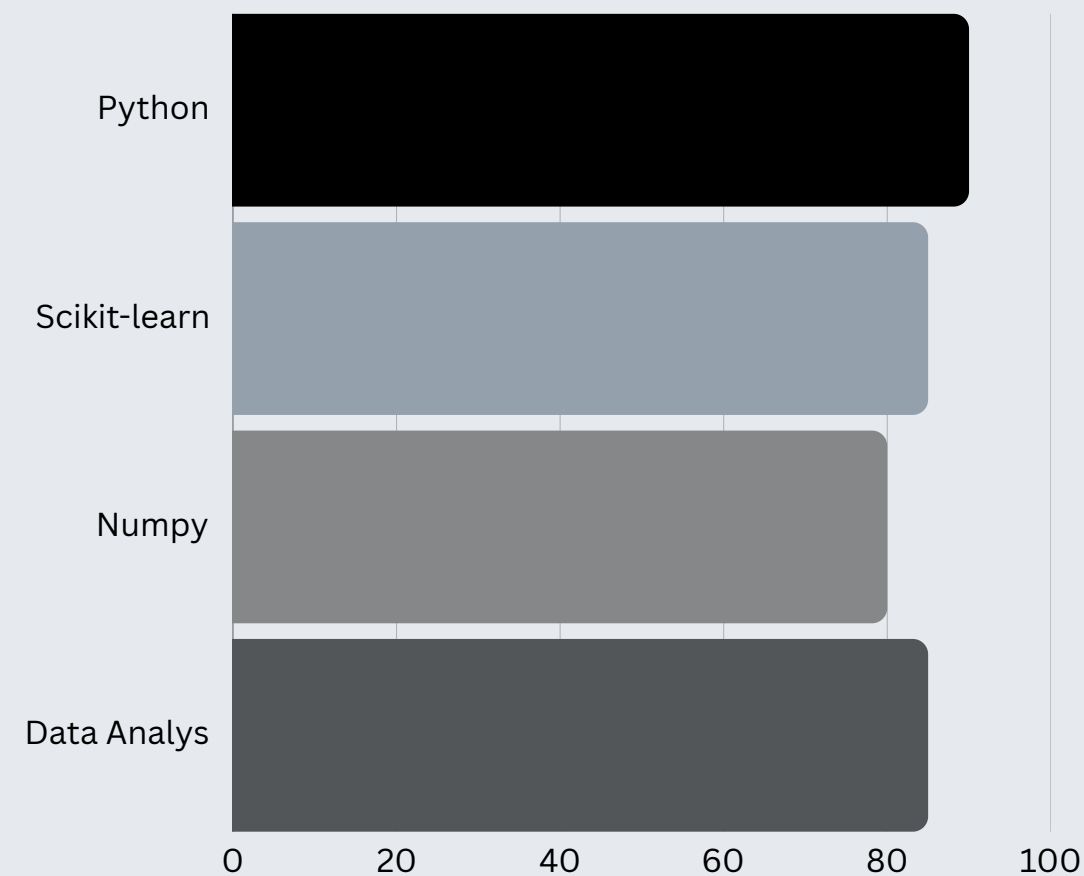
Technical Implementation

★ Data Processing

- MNIST Digits dataset
- 1797 samples
- 8x8 pixel images
- 80/20 train-test split

★ Model Configuration

- KNN Algorithm
- k=5 neighbors
- Euclidean distance metric
- Cross-validation



RESULTS

Project Results



★ Performance Metrics

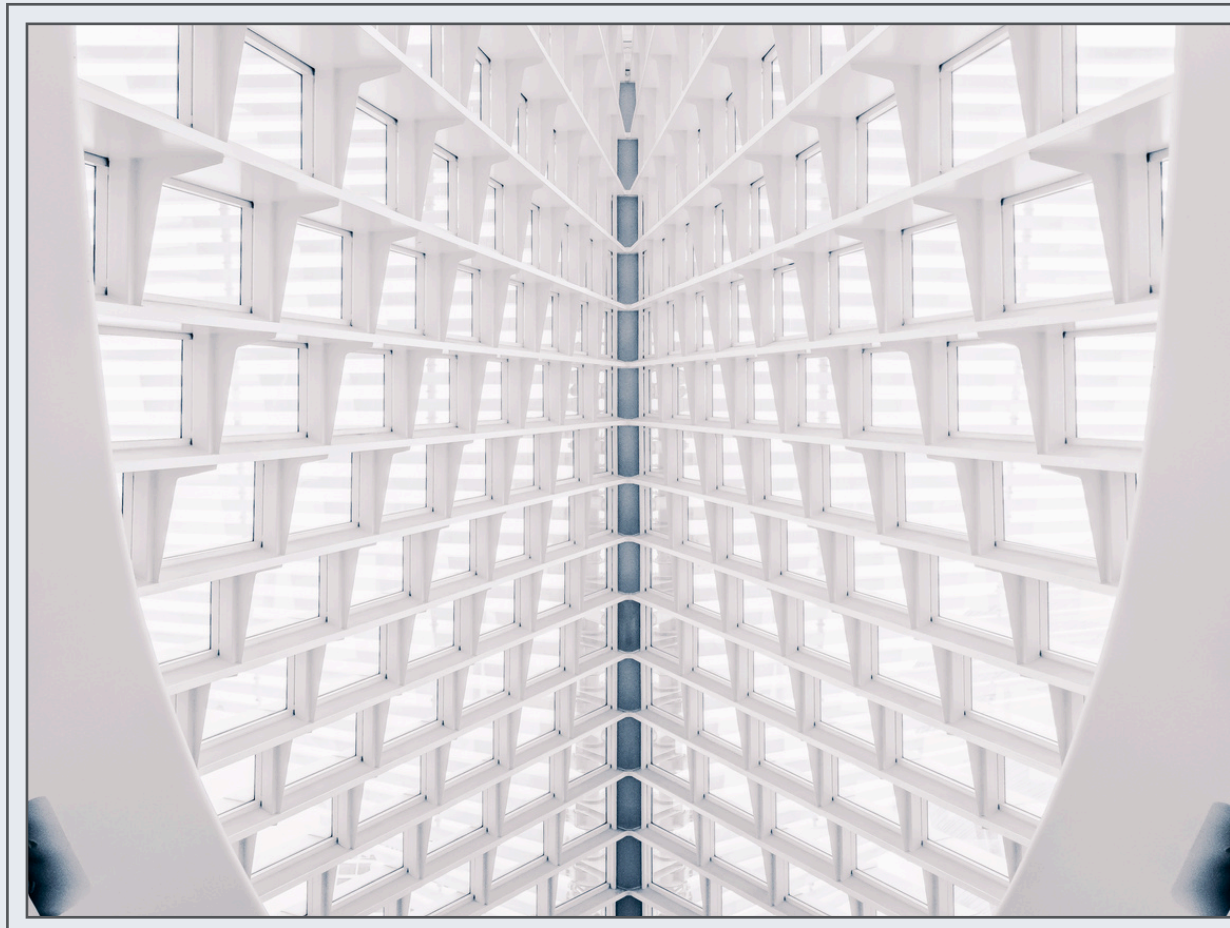
Accuracy	98%
Precision	97%
Recall	96%
F1 Score	97%

★ Key Achievements

- High accuracy across all classes
- Fast training time
- Robust to different handwriting styles
- Minimal false positives

LEARNINGS

Learning Outcomes



★ Technical Skills Gained

- KNN algorithm implementation
- Data preprocessing techniques
- Model evaluation methods
- Performance optimization

★ Future Improvements

- GUI implementation
- Advanced preprocessing
- Model deployment
- Real-time processing

DOCUMENTASI

Project Digit Classification using KNN

```
# Import library yang diperlukan untuk project
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np

# Load dataset digits dari scikit-learn
digits = load_digits()
X = digits.data
y = digits.target

# Membagi data menjadi training dan testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Inisialisasi dan melatih model KNN
# Menggunakan k=5 (5 tetangga terdekat)
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)
```

```
# Melakukan prediksi
y_pred = knn_model.predict(X_test)
```

```
# Evaluasi model
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	33
1	1.00	1.00	1.00	28
2	1.00	1.00	1.00	33
3	1.00	1.00	1.00	34
4	0.98	1.00	0.99	46
5	0.98	0.96	0.97	47
6	0.97	1.00	0.99	35
7	1.00	0.97	0.99	34
8	1.00	1.00	1.00	30
9	0.95	0.95	0.95	40
accuracy			0.99	360
macro avg	0.99	0.99	0.99	360
weighted avg	0.99	0.99	0.99	360

```
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

```
Confusion Matrix:
[[33  0  0  0  0  0  0  0  0  0]
 [ 0 28  0  0  0  0  0  0  0  0]
 [ 0  0 33  0  0  0  0  0  0  0]
 [ 0  0  0 34  0  0  0  0  0  0]
 [ 0  0  0  0 46  0  0  0  0  0]
 [ 0  0  0  0  0 45  1  0  0  1]
 [ 0  0  0  0  0  0 35  0  0  0]
 [ 0  0  0  0  0  0  0 33  0  1]
 [ 0  0  0  0  0  0  0  0 30  0]
 [ 0  0  0  0  1  1  0  0  0 38]]
```

```
# Contoh prediksi untuk satu sampel
sample = X_test[0].reshape(1, -1)
prediction = knn_model.predict(sample)
print(f"\nPrediksi untuk sampel pertama: {prediction[0]}")
print(f"Nilai sebenarnya: {y_test[0]}")
```

```
Prediksi untuk sampel pertama: 6
Nilai sebenarnya: 6
```

```
# Menampilkan akurasi model
accuracy = knn_model.score(X_test, y_test)
print(f"\nAkurasi model: {accuracy:.2f}")
```

```
Akurasi model: 0.99
```

```
# Mencoba beberapa nilai k yang berbeda
k_values = [3, 5, 7, 9, 11]
print("\nPerbandingan akurasi untuk berbagai nilai k:")
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train, y_train)
    score = knn.score(X_test, y_test)
    print(f"k={k}: {score:.4f}")
```

```
Perbandingan akurasi untuk berbagai nilai k:
k=3: 0.9833
k=5: 0.9861
k=7: 0.9889
k=9: 0.9806
k=11: 0.9833
```


Terima Kasih

silviafarida16@gmail.com