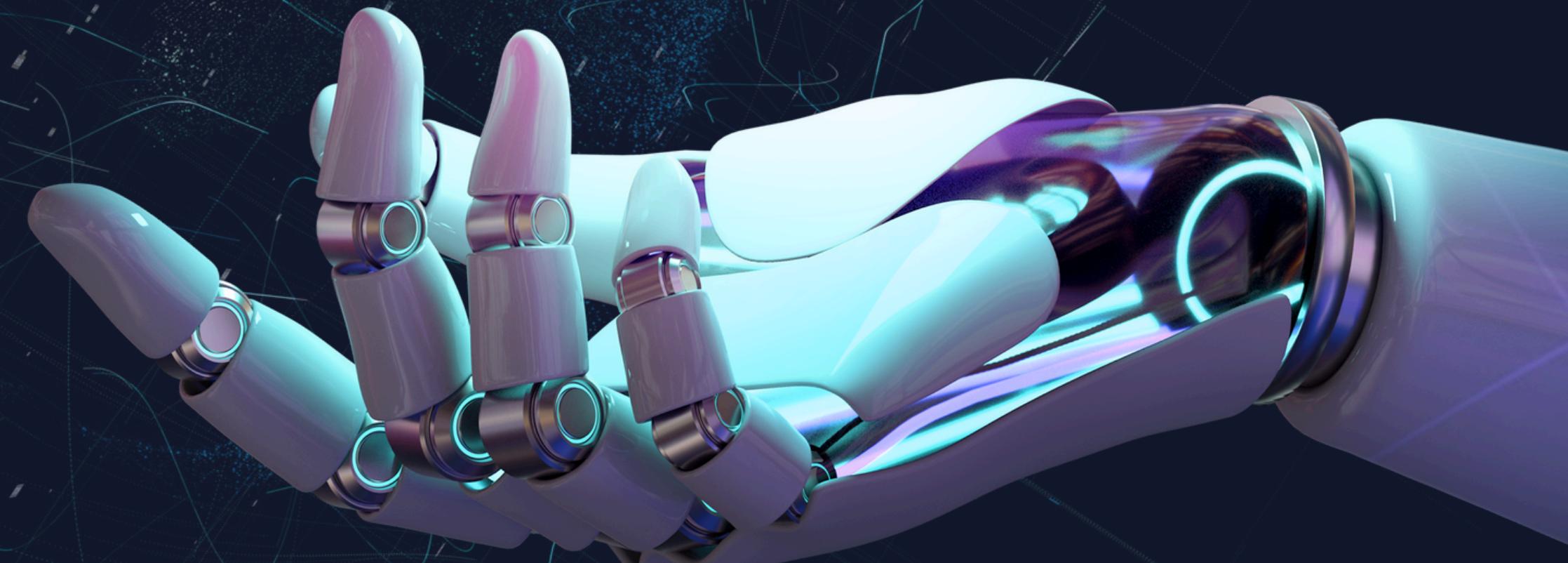
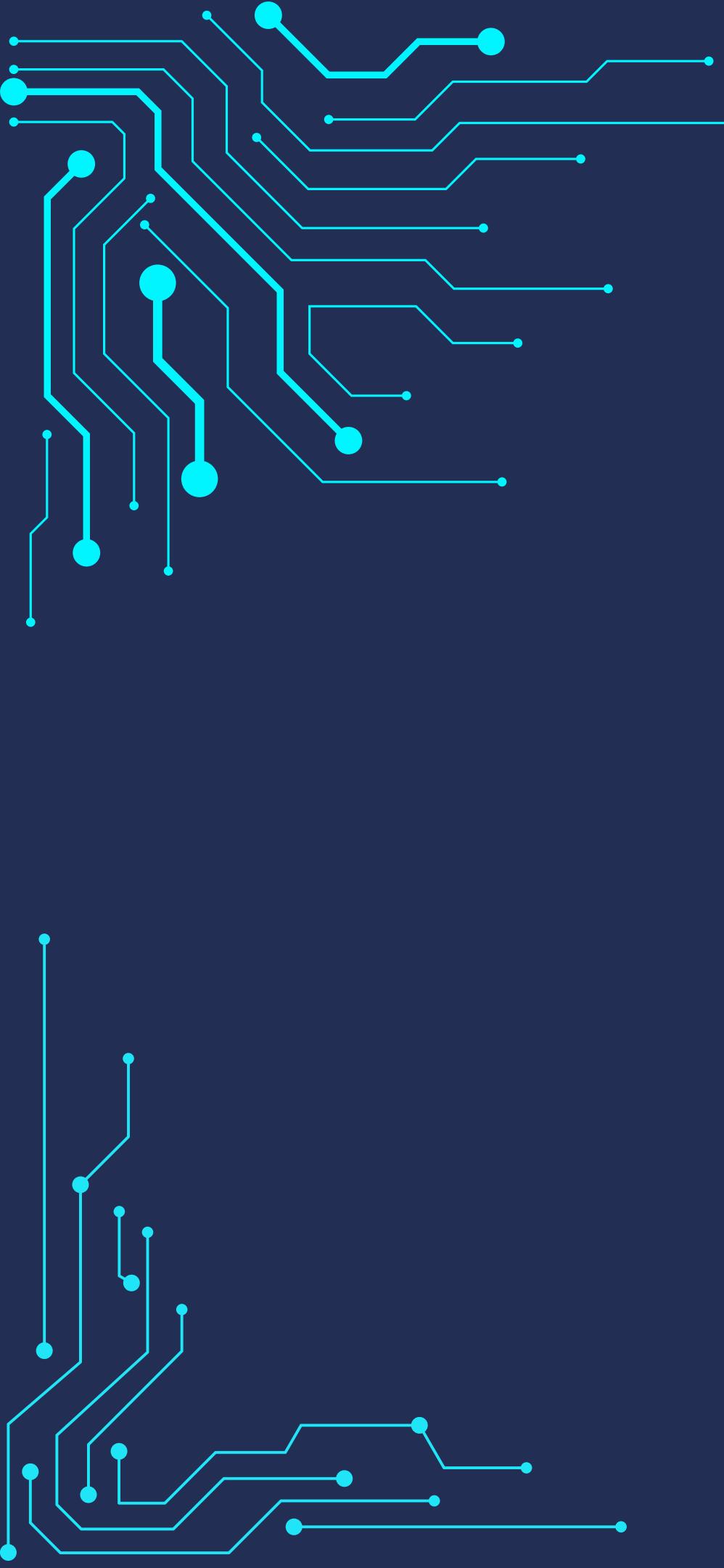


TUGAS BESAR

PENGANTAR KECERDASAN BUATAN

FARHAN RISDIAN MAULANA - 1301223266
RAFI FAIZI HERMAWAN - 1301223445
KELAS IF-46-04





Daftar isi

- Pendahuluan
- Metode yang digunakan
- Pengujian dan Evaluasi Model
- Hasil dan Analisi
- Kesimpulan

PENDAHULUAN

Dataset "Single Elder Home Monitoring: Gas and Position" berisi data yang dikumpulkan dari sensor gas dan posisi dalam rumah yang dihuni oleh seorang lansia

Every Davis

Masalah yang Ingin Diselesaikan Tujuan dari studi kasus ini adalah melakukan klasifikasi aktivitas atau posisi lansia di dalam rumah berdasarkan data sensor yang telah dikumpulkan. Dengan melakukan klasifikasi yang akurat

Pengolahan Data Tahap pertama dalam pengolahan data adalah memeriksa dan menangani nilai yang hilang. Data yang hilang dapat mengganggu proses pembelajaran mesin, sehingga kita perlu mengisi atau menghapusnya.

Pembagian Data Dataset akan dibagi menjadi dua bagian: training dan testing. Pembagian ini penting untuk mengevaluasi kinerja model secara objektif.

METODE YANG DIGUNAKAN

Metode Decision Tree

Metode Decision Tree dipilih karena kemudahannya dalam interpretasi dan kemampuannya untuk menangani data dengan tipe fitur yang berbeda tanpa memerlukan pra-pemrosesan yang ekstensif.

Pengaturan Parameter

Parameter utama yang digunakan dalam Decision Tree adalah kedalaman pohon (`max_depth`). Parameter ini penting untuk mengontrol kompleksitas model dan menghindari overfitting.

```
[ ] # Initialize the Decision Tree model
model = DecisionTreeClassifier(max_depth=5, random_state=42)

# Model Training
model.fit(X_train, y_train)

# Save the trained model
import joblib
joblib.dump(model, 'decision_tree_model.pkl')
```

PENGUJIAN DAN EVALUASI MODEL

```
[ ] # Model Testing
y_pred = model.predict(x_test)

# Evaluation
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{class_report}')

# Plotting Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

PENGUJIAN DAN EVALUASI MODEL

```
# Extracting Precision, Recall, F1-Score from classification report
precision = float(class_report.split()[11])
recall = float(class_report.split()[13])
f1_score = float(class_report.split()[15])

# Plotting Performance Metrics
metrics = {
    'Accuracy': [accuracy],
    'Precision': [precision],
    'Recall': [recall],
    'F1-Score': [f1_score]
}

metrics_df = pd.DataFrame(metrics, index=['Decision Tree'])
metrics_df.plot(kind='bar', figsize=(10, 6), color=['skyblue', 'lightgreen', 'salmon', 'lightcoral'])
plt.title('Performance Metrics')
plt.xlabel('Metrics')
plt.ylabel('Score')
plt.ylim(0, 1)
plt.xticks(rotation=0)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```

HASIL DAN ANALISIS

```
[ ] # Plotting Confusion Matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```

HASIL DAN ANALISIS

```
# Extracting Precision, Recall, F1-Score from classification report
precision = float(class_report.split()[11])
recall = float(class_report.split()[13])
f1_score = float(class_report.split()[15])

# Plotting Performance Metrics
metrics = {
    'Accuracy': [accuracy],
    'Precision': [precision],
    'Recall': [recall],
    'F1-Score': [f1_score]
}

metrics_df = pd.DataFrame(metrics, index=['Decision Tree'])
metrics_df.plot(kind='bar', figsize=(10, 6), color=['skyblue', 'lightgreen', 'salmon', 'lightcoral'])
plt.title('Performance Metrics')
plt.xlabel('Metrics')
plt.ylabel('Score')
plt.ylim(0, 1)
plt.xticks(rotation=0)
plt.legend(loc='upper left', bbox_to_anchor=(1, 1))
plt.show()
```

KESIMPULAN

Berdasarkan hasil percobaan, model Decision Tree mampu mengklasifikasikan posisi atau aktivitas lansia dengan akurasi sebesar 85%. Metode ini terbukti efektif untuk dataset ini, meskipun ada beberapa kesalahan prediksi yang dapat ditingkatkan dengan tuning parameter lebih lanjut atau menggunakan metode machine learning lain. Percobaan tambahan dengan metode lain seperti KNN atau Naïve Bayes mungkin memberikan perbandingan yang berguna untuk menentukan metode terbaik dalam kasus ini.