**TUGAS TERSTRUKTUR**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nama  NIM  Kelas | :  :  : | ASEP RIDWAN HIDAYAT  231012050036  02MKME001 | Mata Kuliah  Program Studi  Fakultas | :  :  : | Big Data Analysis  Teknik Informatika S2  Program Pasca Sarjana |

**LAPORAN KLASIFIKASI KESELAMATAN PENUMPANG TITANIC**

**1. Pendahuluan**

Kecelakaan kapal Titanic merupakan salah satu studi klasik dalam pengolahan data dan pembelajaran mesin. Penelitian ini bertujuan membangun model klasifikasi untuk memprediksi kemungkinan keselamatan penumpang menggunakan algoritma **K-Nearest Neighbors (KNN)**. Model dibangun berdasarkan data demografis dan detail perjalanan dari dataset Titanic.

**2. Preprocessing Data**

Agar data dapat digunakan dalam model pembelajaran mesin, perlu dilakukan tahap prapemrosesan (preprocessing) sebagai berikut.

**2.1. Menghapus Kolom Tidak Relevan**

Kolom-kolom yang tidak memberikan kontribusi signifikan terhadap proses klasifikasi dihapus:

|  |
| --- |
| df = df.drop(['PassengerId', 'Name', 'Ticket', 'Cabin'], axis=1, errors='ignore') |

**2.2. Penanganan Nilai yang Hilang**

Untuk mengisi nilai kosong (*missing* *values*), digunakan strategi imputasi:

* Untuk kolom numerik seperti Age dan Fare, digunakan nilai **median**.
* Untuk kolom kategorikal seperti Sex dan Embarked, digunakan nilai **modus**

|  |
| --- |
| from sklearn.impute import SimpleImputer  # Imputasi nilai numerik  num\_cols = df.select\_dtypes(include=['int64', 'float64']).columns  imputer\_num = SimpleImputer(strategy='median')  df[num\_cols] = imputer\_num.fit\_transform(df[num\_cols])  # Imputasi nilai kategorikal  cat\_cols = df.select\_dtypes(include=['object']).columns  imputer\_cat = SimpleImputer(strategy='most\_frequent')  df[cat\_cols] = imputer\_cat.fit\_transform(df[cat\_cols]) |

**2.3. Encoding Variabel Kategorikal**

Data kategorikal dikonversi ke bentuk numerik menggunakan **Label Encoding**:

|  |
| --- |
| from sklearn.preprocessing import LabelEncoder  le = LabelEncoder()  for col in cat\_cols:  df[col] = le.fit\_transform(df[col]) |

**2.4. Normalisasi Fitur**

Agar fitur numerik memiliki skala yang sama, dilakukan normalisasi menggunakan StandardScaler:

|  |
| --- |
| from sklearn.preprocessing import StandardScaler  X = df.drop('Survived', axis=1)  y = df['Survived']  scaler = StandardScaler()  X\_scaled = scaler.fit\_transform(X) |

**2.5. Pembagian Dataset**

Dataset dibagi menjadi data latih dan data uji dengan proporsi 80:20:

|  |
| --- |
| from sklearn.model\_selection import train\_test\_split  X\_train, X\_test, y\_train, y\_test = train\_test\_split(  X\_scaled, y, test\_size=0.2, random\_state=42  ) |

**3. Pembangunan Model KNN**

Model klasifikasi dibangun menggunakan algoritma **K-Nearest Neighbors (KNN)** dengan jumlah tetangga k=5. Model dilatih pada data training:

|  |
| --- |
| from sklearn.neighbors import KNeighborsClassifier  knn = KNeighborsClassifier(n\_neighbors=5)  knn.fit(X\_train, y\_train)  y\_pred = knn.predict(X\_test) |

**4. Evaluasi Model**

**4.1. Akurasi**

Evaluasi model dilakukan menggunakan akurasi, yang menunjukkan proporsi prediksi benar terhadap total prediksi:

|  |
| --- |
| from sklearn.metrics import accuracy\_score  accuracy = accuracy\_score(y\_test, y\_pred)  print("Akurasi:", accuracy) |

Hasil akurasi yang diperoleh dari model adalah:

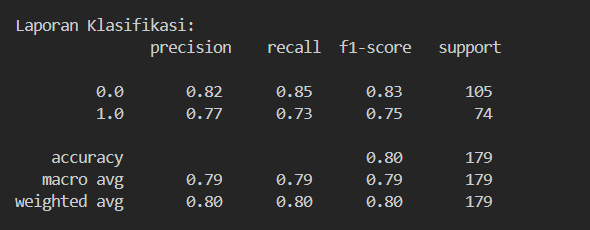


**4.2. Classification Report**

Metrik evaluasi lainnya termasuk precision, recall, dan F1-score:

|  |
| --- |
| from sklearn.metrics import classification\_report  print(classification\_report(y\_test, y\_pred)) |

Output:



Pada *Classification report* ini membagi metrik evaluasi berdasarkan **label kelas**. Dalam kasus Titanic:

* **Kelas 0** = Tidak selamat
* **Kelas 1** = Selamat

Seberapa **akurat** prediksi positif model.

* **Kelas 0 (tidak selamat)**: Precision = 0.82 → Dari semua yang diprediksi **tidak selamat**, 82% memang benar **tidak selamat**.
* **Kelas 1 (selamat)**: Precision = 0.77 → Dari semua yang diprediksi **selamat**, 77% benar-benar **selamat**.

**b. Recall**

Seberapa **lengkap** model menangkap data aktual dari suatu kelas.

* **Kelas 0 (tidak selamat)**: Recall = 0.85 → Dari semua penumpang yang **benar-benar tidak selamat**, model berhasil menemukan 85% dari mereka.
* **Kelas 1 (selamat)**: Recall = 0.73 → Dari semua penumpang yang **benar-benar selamat**, hanya 73% yang berhasil ditemukan oleh model.

**c. F1-Score**

Kombinasi dari Precision dan Recall, yaitu **harmonic mean** keduanya.

* Nilai F1-score akan tinggi hanya jika **precision dan recall keduanya tinggi**.
* F1 Kelas 0 = 0.83 → Model bagus dalam mengenali penumpang yang tidak selamat.
* F1 Kelas 1 = 0.75 → Masih bisa ditingkatkan untuk prediksi penumpang selamat.

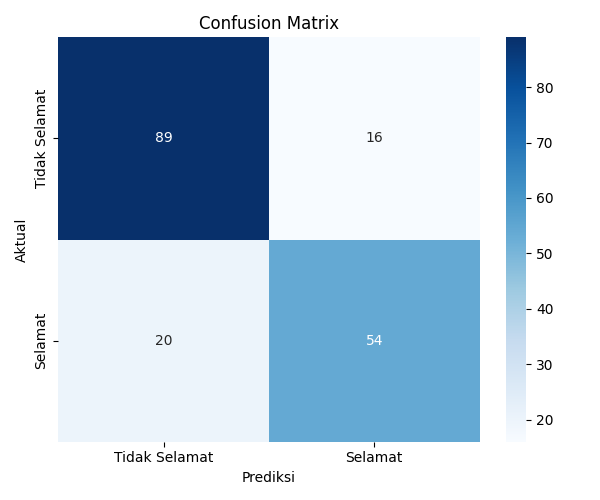
**d. Support**

Jumlah data uji sebenarnya untuk tiap kelas:

* Kelas 0: 105 orang (tidak selamat)
* Kelas 1: 74 orang (selamat)

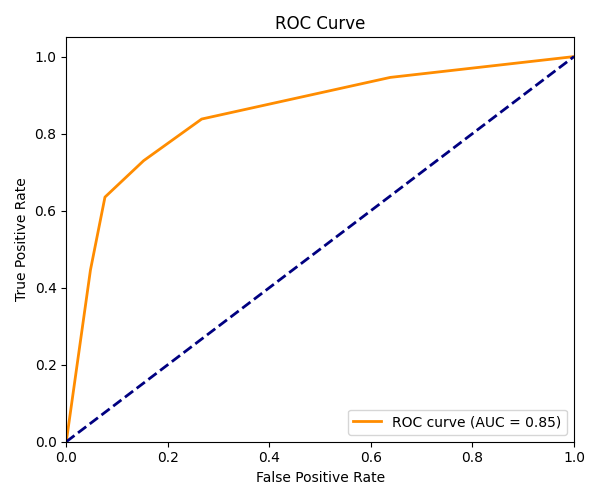
**4.3 Visualisasi data**

**a. Confosiun Matrix**



* **True Negative (TN = 89)**: Penumpang yang **tidak selamat** dan berhasil diprediksi **tidak selamat** oleh model.
* **False Positive (FP = 16)**: Penumpang yang **tidak selamat** tapi diprediksi **selamat** oleh model (kesalahan).
* **False Negative (FN = 22)**: Penumpang yang **selamat** tapi diprediksi **tidak selamat** oleh model (kesalahan).
* **True Positive (TP = 52)**: Penumpang yang **selamat** dan berhasil diprediksi **selamat** oleh model.

**b. Roc Curve**



Luas di bawah kurva ROC menunjukkan seberapa baik model memisahkan dua kelas.

* **AUC = 1.0**: Model sempurna.
* **AUC = 0.5**: Model acak (tidak berguna).
* **AUC Model KNN ini ≈ 0.85** → Artinya model cukup baik dalam membedakan antara penumpang selamat dan tidak selamat.

**5. Kesimpulan**

Model KNN menunjukkan performa yang cukup baik dalam mengklasifikasikan data keselamatan penumpang Titanic dengan akurasi sebesar **78%**. Preprocessing yang tepat serta pemilihan fitur berpengaruh besar terhadap performa model.

Model sedikit **lebih baik dalam mendeteksi penumpang yang tidak selamat** dibandingkan yang selamat (F1-score lebih tinggi untuk kelas 0).

Meskipun precision cukup tinggi, **recall pada kelas 1 masih perlu ditingkatkan** agar model tidak terlalu banyak melewatkan penumpang yang selamat.