UTS Machine Learning - Tugas Klasifikasi

Nama: M Faishal Abdurrahman

NIM: 1103213015

Mount Google Drive

```
from google.colab import drive
_drive_mount_path = '/content/drive'
drive.mount(_drive_mount_path)

# Path ke file CSV di Drive
file_path = f"{_drive_mount_path}/MyDrive/UTS/KlasifikasiUTS.csv"
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import (confusion_matrix, classification_report,
                             roc_auc_score, roc_curve, auc)

# Model
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import BaggingClassifier, AdaBoostClassifier
from sklearn.svm import SVC
# Seleksi fitur
from sklearn.feature_selection import mutual_info_classif
```

Load & Inspeksi Data

```
# Load dataset
df = pd.read_csv(file_path)
# Tampilkan ukuran dan kolom\ nprint("Bentuk data:", df.shape)
print("Daftar kolom:", df.columns.tolist())
df.head()
```

```
Daftar kolom: ['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17',
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.11047
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.10128
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.90941
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.19032
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.13745

5 rows × 31 columns

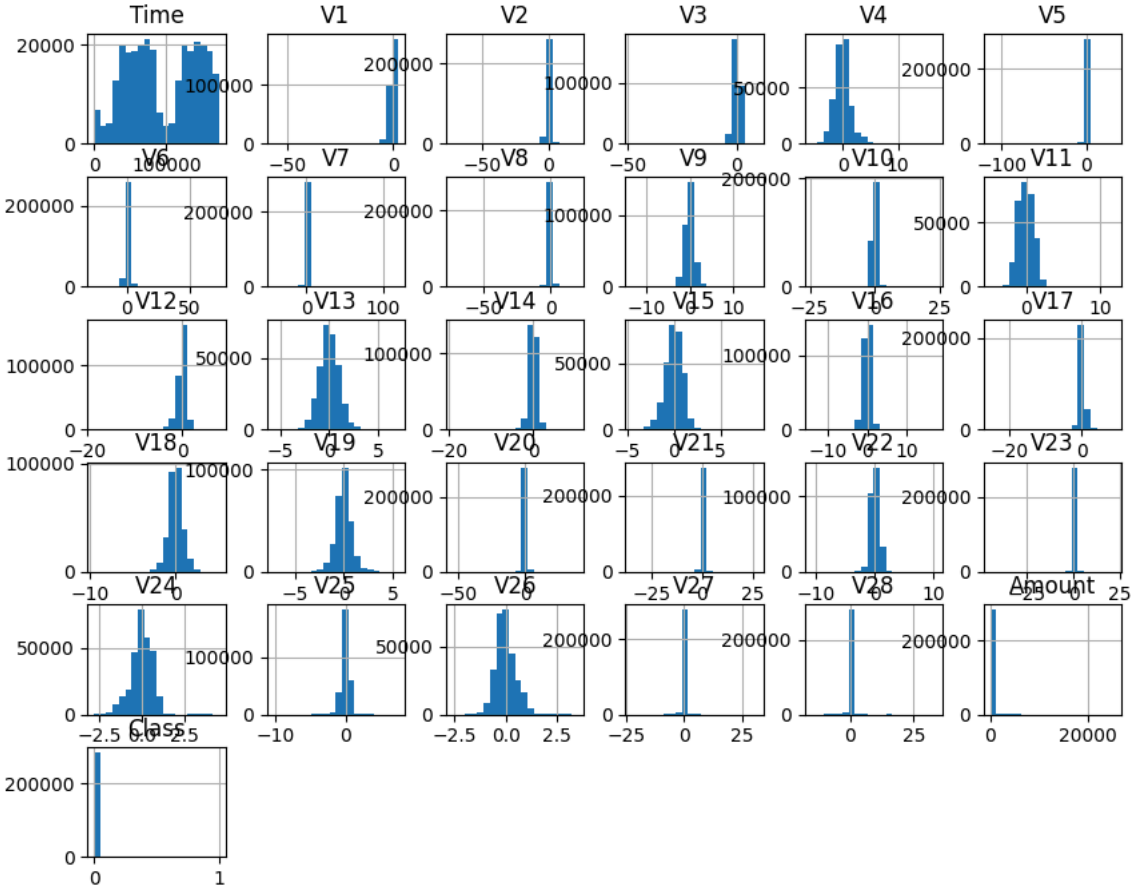
Analisis Data Eksplorasi (EDA)

```
# Cek nilai hilang
print(df.isnull().sum())

# Statistik dasar
print(df.describe(include='all'))

# Proporsi kelas
print(df['Time'].value_counts(normalize=True))
```

```
# Histogram fitur numerik
df.hist(bins=20, figsize=(10,8));
```

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
                Time             V1             V2             V3             V4  \
count  284807.000000  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean    94813.859575  1.168375e-15  3.416908e-16 -1.379537e-15  2.074095e-15
std     47488.145955  1.958696e+00  1.651309e+00  1.516255e+00  1.415869e+00
min         0.000000 -5.640751e+01 -7.271573e+01 -4.832559e+01 -5.683171e+00
25%     54201.500000 -9.203734e-01 -5.985499e-01 -8.903648e-01 -8.486401e-01
50%     84692.000000  1.810880e-02  6.548556e-02  1.798463e-01 -1.984653e-02
75%    139320.500000  1.315642e+00  8.037239e-01  1.027196e+00  7.433413e-01
max    172792.000000  2.454930e+00  2.205773e+01  9.382558e+00  1.687534e+01

                 V5             V6             V7             V8             V9  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   9.604066e-16  1.487313e-15 -5.556467e-16  1.213481e-16 -2.406331e-15
std    1.380247e+00  1.332271e+00  1.237094e+00  1.194353e+00  1.098632e+00
min   -1.137433e+02 -2.616051e+01 -4.355724e+01 -7.321672e+01 -1.343407e+01
25%   -6.915971e-01 -7.682956e-01 -5.540759e-01 -2.086297e-01 -6.430976e-01
50%   -5.433583e-02 -2.741871e-01  4.010308e-02  2.235804e-02 -5.142873e-02
75%    6.119264e-01  3.985649e-01  5.704361e-01  3.273459e-01  5.971390e-01
max    3.480167e+01  7.330163e+01  1.205895e+02  2.000721e+01  1.559499e+01

       ...            V21            V22            V23            V24  \
count  ...  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05
mean   ...  1.654067e-16 -3.568593e-16  2.578648e-16  4.473266e-15
std    ...  7.345240e-01  7.257016e-01  6.244603e-01  6.056471e-01
min    ... -3.483038e+01 -1.093314e+01 -4.480774e+01 -2.836627e+00
25%    ... -2.283949e-01 -5.423504e-01 -1.618463e-01 -3.545861e-01
50%    ... -2.945017e-02  6.781943e-03 -1.119293e-02  4.097606e-02
75%    ...  1.863772e-01  5.285536e-01  1.476421e-01  4.395266e-01
max    ...  2.720284e+01  1.050309e+01  2.252841e+01  4.584549e+00

                V25            V26            V27            V28          Amount  \
count  2.848070e+05  2.848070e+05  2.848070e+05  2.848070e+05  284807.000000
mean   5.340915e-16  1.683437e-15 -3.660091e-16 -1.227390e-16      88.349619
std    5.212781e-01  4.822270e-01  4.036325e-01  3.300833e-01     250.120109
min   -1.029540e+01 -2.604551e+00 -2.256568e+01 -1.543008e+01       0.000000
25%   -3.171451e-01 -3.269839e-01 -7.083953e-02 -5.295979e-02       5.600000
50%    1.659350e-02 -5.213911e-02  1.342146e-03  1.124383e-02      22.000000
75%    3.507156e-01  2.409522e-01  9.104512e-02  7.827995e-02      77.165000
max    7.519589e+00  3.517346e+00  3.161220e+01  3.384781e+01   25691.160000

              Class
count  284807.000000
mean        0.001727
std         0.041527
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000

[8 rows x 31 columns]
Time
163152.0    0.000126
64947.0     0.000091
68780.0     0.000088
3767.0      0.000074
3770.0      0.000070
              ...
```

```
172760.0    0.000004
172758.0    0.000004
172757.0    0.000004
172756.0    0.000004
172754.0    0.000004
Name: proportion, Length: 124592, dtype: float64
```



Pembersihan & Persiapan Data

```
# 5.1 Buang duplikasi
awal = len(df)  # Store initial length of the dataframe
df = df.drop_duplicates()
print(f"Duplikasi yang dihapus: {awal - len(df)} jika ada duplikat sebelumnya")

# 5.2 Imputasi nilai hilang
default_numeric = df.select_dtypes(include=[np.number]).columns
for col in default_numeric:
    df[col] = df[col].fillna(df[col].median())

default_categorical = df.select_dtypes(include=['object','category']).columns
for col in default_categorical:
    df[col] = df[col].fillna(df[col].mode()[0])

# 5.3 Definisikan X dan y
# Ganti 'Time' dengan nama kolom target yang benar jika perlu
X = df.drop('Time', axis=1)
y = df['Time']

# 5.4 Bagi data latih dan uji
# Coba stratify, jika gagal gunakan tanpa stratify
try:
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, stratify=y, random_state=42
    )
    print("Split data dengan stratify berhasil.")
except ValueError:
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )
    print("Perhatian: stratify dihilangkan karena kelas target hanya memiliki satu anggota di salah satu kelas.")
```

```
Duplikasi yang dihapus: 1081 jika ada duplikat sebelumnya
Perhatian: stratify dihilangkan karena kelas target hanya memiliki satu anggota di salah satu kelas.
```

Seleksi Fitur

```
# Subsampling untuk mutual information (misal 5000 baris acak)
sample_n = 5000 if len(X_train) > 5000 else len(X_train)
X_sample = X_train.sample(n=sample_n, random_state=42)
y_sample = y_train.loc[X_sample.index]

# Identifikasi fitur numerik dan kategorikal
numeric_features = X_train.select_dtypes(include=[np.number]).columns.tolist()
categorical_features = X_train.select_dtypes(include=['object','category']).columns.tolist()

# 6.1 Mutual Information pada sampel
from sklearn.feature_selection import mutual_info_classif
nmi = mutual_info_classif(X_sample[numeric_features], y_sample)
mi_series = pd.Series(nmi, index=numeric_features).sort_values(ascending=False)
print("Mutual Information (numerik, sampel):")
print(mi_series)

# 6.2 ANOVA F-test alternatif
from sklearn.feature_selection import SelectKBest, f_classif
selector = SelectKBest(score_func=f_classif, k=5)
selector.fit(X_sample[numeric_features], y_sample)
f_scores = pd.Series(selector.scores_, index=numeric_features).sort_values(ascending=False)
print("ANOVA F-test (numerik, sampel):")
print(f_scores)

# Pilih top-5 fitur berdasarkan MI atau F-test
top_mi = mi_series.head(5).index.tolist()
top_f = f_scores.head(5).index.tolist()
print(f"Fitur terpilih MI: {top_mi}")
print(f"Fitur terpilih F-test: {top_f}")
```

```
Mutual Information (numerik, sampel):
V3        0.276647
V22       0.266096
V28       0.209447
V1        0.205610
V4        0.166494
V25       0.160290
V10       0.158318
V20       0.154535
V21       0.146194
V5        0.105414
```

```
V12       0.103310
V26       0.072283
V8        0.065926
V27       0.063414
V18       0.033739
V7        0.029086
V9        0.024547
V14       0.017451
V23       0.006583
V17       0.006216
V2        0.000000
V6        0.000000
V13       0.000000
V11       0.000000
V16       0.000000
V15       0.000000
V19       0.000000
V24       0.000000
Amount    0.000000
Class     0.000000
dtype: float64
ANOVA F-test (numerik, sampel):
Class          inf
V27       3.812798
V23       3.499538
V21       2.181846
V20       2.058505
V12       2.029267
V8        1.899197
V28       1.617705
V22       1.536174
V3        1.502144
V13       1.455156
V4        1.433041
V5        1.432654
V7        1.407012
V15       1.269991
V14       1.265607
V17       1.251741
V25       1.183785
V11       1.168635
Amount    1.163620
V1        1.154542
V16       1.066302
V9        1.054352
V2        1.051255
```

Pipeline Pra-pemrosesan & Rekayasa Fitur Pipeline Pra-pemrosesan & Rekayasa Fitur

```
# Tentukan fitur terpilih
top_features = top_mi  # atau top_f
print(f"Menggunakan fitur terpilih: {top_features}")

# Subset data latih dan uji
X_train = X_train[top_features]
X_test  = X_test[top_features]

# Buat pipeline hanya untuk fitur terpilih
after_num = Pipeline([('scaler', StandardScaler())])
# Jika ada kategorikal, buat cat pipeline
after_cat = Pipeline([('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=True))])

# Preprocessor sekarang hanya untuk numeric
preprocessor = ColumnTransformer([
    ('num', after_num, top_features),
    # ('cat', after_cat, categorical_features)
])
```

```
Menggunakan fitur terpilih: ['V3', 'V22', 'V28', 'V1', 'V4']
```

Fungsi Pelatihan & Evaluasi Model

```
from sklearn.model_selection import cross_val_score

def buat_model(seed=42):
    return {
        'LogisticRegression': LogisticRegression(max_iter=1000, random_state=seed),
        'DecisionTree': DecisionTreeClassifier(random_state=seed),
        'KNN': KNeighborsClassifier(),
        'Bagging': BaggingClassifier(random_state=seed),
        'AdaBoost': AdaBoostClassifier(random_state=seed),
        'SVM': SVC(probability=True, random_state=seed)
    }
```