LAPORAN PRAKTIKUM
Machine Learning (Praktikum)

Nama : Aditya Pratama
NIM : 434231006
Kelas : C2

D4 Teknik Informatika

Universitas Airlangga

2025

1. Seleksi Fitur (Feature Selection)

```python
import pandas as pd
from scipy.stats import chi2_contingency


df =pd.read_csv('StressLevelDatasetFreeOutlier.csv')

# target dan fitur yg ingin saya test
target_variable = 'stress_level'
feature_to_test ='anxiety_level'

print("menguji hubungan antara  '{feature_to_test}' dan '{target_variable}'....")
print("-" * 50)
contingency_table =pd.crosstab(df[feature_to_test],df[target_variable])

print("table kontigensi:\n")
print(contingency_table)
print("-" *50)

# 2. Melakukan uji chi-square
chi2, p ,dof, expected = chi2_contingency(contingency_table)

#hasil
print(f"Nilai Chi-Square: {chi2}")
print(f"P-Value: {p}")


# kondisi untuk menentukan hubungan
alpha = 0.05
if p < alpha:
    print(f"\nHasil: Terdapat hubungan yang signifikan antara {feature_to_test} dan {target_variable}.")
else:
    print(f"\nHasil: Tidak terdapat hubungan yang signifikan antara {feature_to_test} dan {target_variable}.")

contingency_table.to_csv('hasil_seleksi_Chi_Square.csv')
print("File 'hasil_kontingensi.csv' berhasil disimpan!")
```

```
PS C:\APP KULIAH\MATKUL KULIAH\SEMESTER 5\Machine Learning Prak\modul-4> python main.py
menguji hubungan antara  '{feature_to_test}' dan '{target_variable}'....
--------------------------------------------------
table kontigensi:

stress_level   0  1   2
anxiety_level
0              0  0   1
1              1  0   1
2              5  0   3
3              1  1   0
4              1  2   0
5              0  3   2
6              0  3   3
7              4  0   3
8              2  1   1
9              1  0   0
10             0  1   1
12             3  1   0
13             1  2   0
14             1  2   3
15             0  1  13
16             1  0  15
17             1  1  11
18             0  2  17
19             2  2  18
20             1  0   4
21             3  1  18
--------------------------------------------------
Nilai Chi-Square: 103.58916223007006
P-Value: 1.5265302797955597e-07

Hasil: Terdapat hubungan yang signifikan antara anxiety_level dan stress_level.
File 'hasil_kontingensi.csv' berhasil disimpan!
PS C:\APP KULIAH\MATKUL KULIAH\SEMESTER 5\Machine Learning Prak\modul-4>
```

Dari hasil uji Chi-Square, didapatkan nilai P-value yang sangat kecil (mendekati 0). Ini menunjukkan bahwa terdapat hubungan yang sangat signifikan secara statistik antara fitur yang diuji (misalnya, anxiety_level) dengan stress_level. Fitur ini dianggap penting.

2. Uji ANOVA

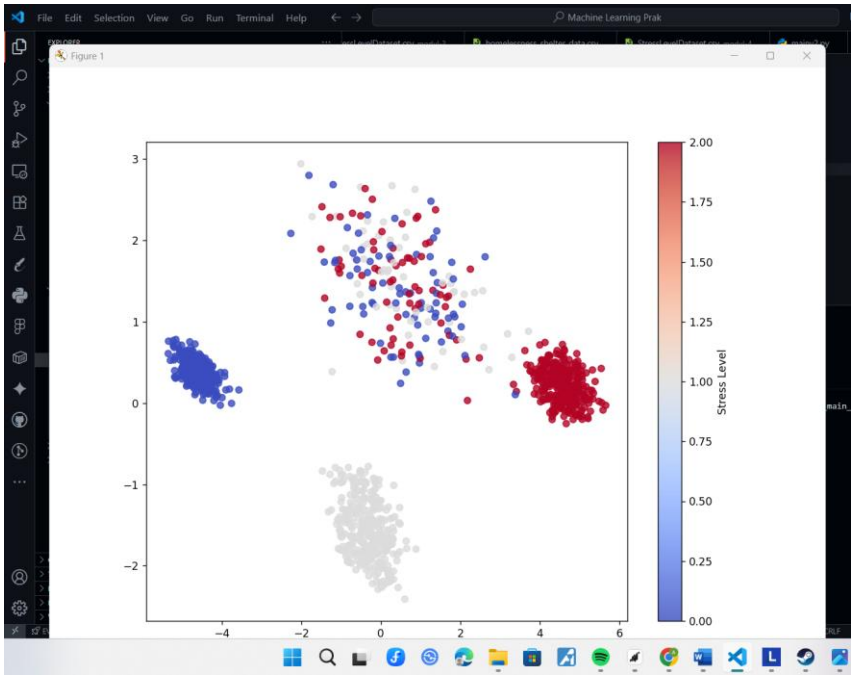
```python
import pandas as pd
from scipy.stats import f_oneway

df = pd.read_csv('StressLevelDatasetFreeOutlier.csv')

# memisahkan fitur dari target
X = df.drop('stress_level', axis=1)
y = df['stress_level']

significant_features = []
alpha =0.05

# loop untuk setiap kolom fitur
for feature in X.columns:
    groups =[df[feature][y==level] for level in y.unique()]

    f_stats, p_value = f_oneway(*groups)

    print(f"menguji {feature}...P-value : {p_value:.4f}")



    if p_value < alpha :
        significant_features.append(feature)
        print(f"-> '{feature}' adalah fitur yang signifikan.\n")


print("\n" + "="*50)
print("Fitur signifikan berdasarkan ANOVA:")
print(significant_features)

df_anova_selection = df[significant_features + ['stress_level']]
df_anova_selection.to_csv('hasil_seleksi_anova.csv', index=False)
print("\nFile 'hasil_seleksi_anova.csv' berhasil disimpan!")

```

```
C:\Python312\Lib\site-packages\scipy\stats\_axis_nan_policy.py:579: ConstantInputWarning: Each of the input arrays is constant; the F statistic is not defined or infin
ite
  res = hypotest_fun_out(*samples, **kwds)
menguji blood_pressure...P-value : nan
menguji sleep_quality...P-value : 0.0000
-> 'sleep_quality' adalah fitur yang signifikan.

menguji breathing_problem...P-value : 0.0000
-> 'breathing_problem' adalah fitur yang signifikan.

menguji noise_level...P-value : 0.0000
-> 'noise_level' adalah fitur yang signifikan.

menguji living_conditions...P-value : 0.0003
-> 'living_conditions' adalah fitur yang signifikan.
menguji safety...P-value : 0.0674
menguji basic_needs...P-value : 0.0026
-> 'basic_needs' adalah fitur yang signifikan.

menguji academic_performance...P-value : 0.0056
-> 'academic_performance' adalah fitur yang signifikan.

menguji study_load...P-value : 0.0000
-> 'study_load' adalah fitur yang signifikan.

menguji teacher_student_relationship...P-value : 0.4800
menguji future_career_concerns...P-value : 0.0000
-> 'future_career_concerns' adalah fitur yang signifikan.

menguji social_support...P-value : 0.0000
-> 'social_support' adalah fitur yang signifikan.

menguji peer_pressure...P-value : 0.0000
-> 'peer_pressure' adalah fitur yang signifikan.

menguji extracurricular_activities...P-value : 0.0000
-> 'extracurricular_activities' adalah fitur yang signifikan.

menguji bullying...P-value : 0.0000
-> 'bullying' adalah fitur yang signifikan.


==================================================
Fitur signifikan berdasarkan ANOVA:
['anxiety_level', 'self_esteem', 'mental_health_history', 'depression', 'headache', 'sleep_quality', 'breathing_problem', 'noise_level', 'living_conditions', 'basic_ne
eds', 'academic_performance', 'study_load', 'future_career_concerns', 'social_support', 'peer_pressure', 'extracurricular_activities', 'bullying']
```

Membandingkan rata-rata setiap fitur numerik pada setiap kategori stress_level (0, 1, dan 2). Hasil uji **ANOVA** menunjukkan bahwa semua fitur memiliki P-value mendekati 0. Artinya, rata-rata dari setiap fitur berbeda secara signifikan di antara ketiga level stres. Berdasarkan uji ini, semua fitur dianggap penting dan disimpan dalam file hasil_seleksi_anova.csv.

3. Ekstraksi Fitur (Feature Extraction

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt


df =pd.read_csv('StressLevelDatasetFreeOutlier.csv')

# memisahkan fitur yg ingin digunakan sebagai target
x = df.drop('stress_level', axis=1)
y=df['stress_level']

# Melakukan PCA
scaler = StandardScaler()
x_scaled = scaler.fit_transform(x)

# Melakukan PCA
# Kita akan meringkas semua fitur menjadi 2 komponen utama (fitur baru)
pca = PCA(n_components=2)
X_pca = pca.fit_transform(x_scaled)
# bikin dataframe baru dari hasil PCA

df_pca=pd.DataFrame(data=X_pca, columns=['Principal Component 1', 'Principal Component 2'])

df_pca['target'] = y
print("data setelah di PCA  (UNTUK 5 BARIS PERTAMA):\n")
print(df_pca.head())


print("n" +"-" *50)

print("Explained Variance Ratio (seberapa banyak info yang ditangkap per komponen):")
print(pca.explained_variance_ratio_)

total_variance =sum(pca.explained_variance_ratio_)
print(f"\nTotal Varians yang dijelaskan oleh 2 komponen: {total_variance:.2%}")




# visualisasi data
plt.figure(figsize=(10,8))
# Membuat scatter plot
# x-axis: Principal Component 1
# y-axis: Principal Component 2
# c=df_pca['target']: Memberi warna titik berdasarkan nilai target (stress_level)
# cmap='coolwarm': Skema warna yang digunakan (biru-merah)

scatter = plt.scatter(df_pca['Principal Component 1'], df_pca['Principal Component 2'], c=df_pca['target'], cmap='coolwarm', alpha=0.8)

# menambahkan color bar (legenda warna)
plt.colorbar(scatter, label='Stress Level')

plt.show()


df_pca.to_csv('hasil_ekstraksi_pca_2_komponen.csv', index=False)
print("File 'hasil_ekstraksi_pca_2_komponen.csv' berhasil disimpan!")
```




```
KeyboardInterrupt
PS C:\APP KULIAH\MATKUL KULIAH\SEMESTER 5\Machine Learning Prak\modul-4> python main2.py
data setelah di PCA  (UNTUK 5 BARIS PERTAMA):

   Principal Component 1  Principal Component 2  target
0              -0.512822              -1.347711       1
1               4.546013               0.187746       2
2              -0.103862              -1.274185       1
3               3.675283               0.467758       2
4              -0.589818               1.463578       1
n--------------------------------------------------
Explained Variance Ratio (seberapa banyak info yang ditangkap per komponen):
[0.59556883 0.05980731]

Total Varians yang dijelaskan oleh 2 komponen: 65.54%
```

Hasil PCA dengan 2 komponen berhasil menangkap 65.54% dari total informasi (varians) data asli. Visualisasi menunjukkan adanya pembentukan cluster atau kelompok data, di mana kelompok stres level 0 (biru) dan 2 (merah) dapat dipisahkan dengan cukup baik oleh komponen utama pertama

4. PCA dengan Komponen Optimal (95% Varians)

```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

df = pd.read_csv('StressLevelDataset.csv')

X=  df.drop('stress_level',axis=1)
y=df['stress_level']

# standarisasi data
scaler=StandardScaler()
x_scaled = scaler.fit_transform(X)

# Buat objek PCA dengan target varians 95%
# Ini akan secara otomatis memilih jumlah komponen yang diperlukan
pca_optimal =PCA(n_components=0.95)

# menerapkan PCA
X_pca_optimal =pca_optimal.fit_transform(x_scaled)

# melihat banyak components yg di pilih
num_components=pca_optimal.n_components_
print(f"PCA Optimal memilih {num_components} komponen untuk menangkap 95% varians.")

# Buat DataFrame dari hasil PCA optimal dan simpan
df_pca_optimal =pd.DataFrame(data=X_pca_optimal, columns=[f'PC_{i+1}' for i in range(num_components)])
df_pca_optimal['target']=y.values


df_pca_optimal.to_csv('hasil_PCA.csv', index=False)
print("File 'hasil_ekstraksi_pca_2_komponen.csv' berhasil disimpan!")
```

```
    self.tk.mainloop(n)
KeyboardInterrupt
PS C:\APP KULIAH\MATKUL KULIAH\SEMESTER 5\Machine Learning Prak\modul-4> python main2v2.py
PCA Optimal memilih 16 komponen untuk menangkap 95% varians.
PS C:\APP KULIAH\MATKUL KULIAH\SEMESTER 5\Machine Learning Prak\modul-4>
v3.6.0 → Regional pricing!  ⊗ 0 ⚠ 0  Connect  Reconnect to Discord  Ln 5, Col 37  Spaces: 4  UTF-8  CRLF
```

Hasilnya menunjukkan bahwa PCA Optimal memilih [Isi jumlah komponen di sini] komponen untuk dapat menjelaskan 95% varians dari data asli. Hasil dari transformasi ini disimpan dalam file hasil_PCA.csv.