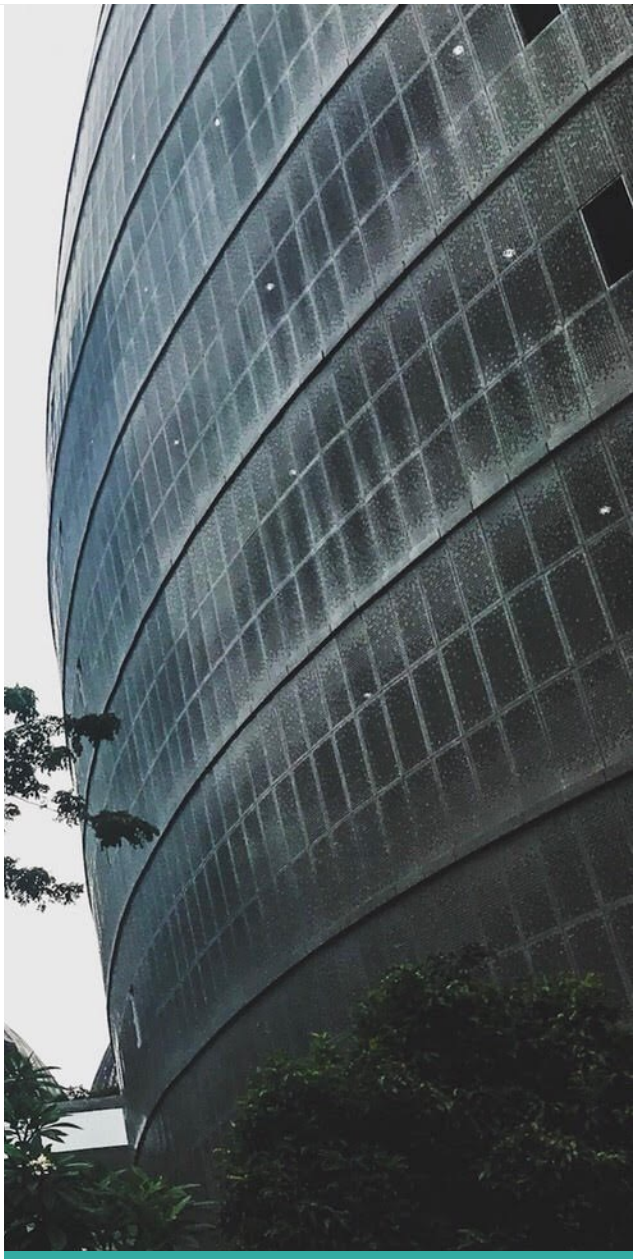
MODUL PRAKTIKUM

IS411 – DATA MODELING
PROGRAM SARJANA S1 SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA



PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK DAN INFORMATIKA
UNIVERSITAS MULTIMEDIA NUSANTARA

Gedung B Lantai 5, Kampus UMN
Jl. Scientia Boulevard, Gading Serpong, Tangerang, Banten-15811 Indonesia
Telp: +62-21.5422.0808 (ext. 1803), email: ict.lab@umn.ac.id, web: umn.ac.id

MODUL 7
INTRODUCTION TO MACHINE LEARNING

DESKRIPSI TEMA

1. Supervised Learning
2. Unsupervised Learning

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Students are able to explain the concept of machine learning (C2)

PENUNJANG PRAKTIKUM

1. Anaconda Navigator
2. Jupyter Notebook

(+ Perlengkapan Apd/Alat Pelindung Diri Yang Harus Digunakan, Jika Ada)

LANGKAH-LANGKAH PRAKTIKUM

Import Library

1. Lakukan langkah yang sama seperti minggu sebelumnya untuk meng-import library Numpy dan Pandas.

Supervised Learning

Supervised Learning dalam machine learning adalah cara sistem mempelajari data dengan bantuan label yang sudah diberikan oleh user sebelumnya. Biasanya supervised learning digunakan untuk melakukan klasifikasi (pengelompokkan) data berdasarkan independent variable.

Berikut ini adalah contoh data redwine dengan label yang sudah diberikan.

Import Data

1. Gunakan dataset **winequality-red.csv** dan masukkan ke dalam dataframe menggunakan Pandas.
2. Lakukan langkah berikut.

```
1 print(dataset.groupby('quality').size())
2 dataset.info()
```

Pada data tersebut, quality merupakan label yang akan dicari oleh sistem, sehingga machine learning yang dibuat nantinya akan membaca karakteristik wine baik dari acid, sugar, sulfur,

density, dan lainnya. Setelah membaca karakteristik wine, barulah sistem menentukan kualitas wine.

3. Langkah awalnya adalah memisahkan kolom independent variable dengan dependent variable.

```
redwine = dataset.copy()
Y = redwine['quality']
X = redwine.drop(columns = 'quality')
```

4. Setelah memisahkan kolom tersebut, bagi data tersebut menjadi 2 bagian, untuk training dan testing menggunakan bantuan library dari SKlearn.

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)

```

Ada beberapa algoritma untuk melakukan klasifikasi data,dan algoritma yang populer digunakan adalah Logistic Regression, Decision Tree Classifier, K Neighbors Classifier dan Gaussian Naive-Bayes.

5. Logistic Regression

Secara sederhana, logistic regression adalah algoritma yang menganalisa data kemudian memberikan output binary seperti "yes" dan "no".

Dalam contoh ini, output yang diberikan cukup beragam, sehingga algoritma ini kurang efektif untuk digunakan. Jika output lebih dari 2, biasanya akan digunakan Multinomial Logistic Regression, dimana pembagian data untuk testing dan training akan menjadi lebih kompleks.

```
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()
logreg.fit(X_train, y_train)
print('Accuracy of Logistic regression classifier on training set: {:.2f}'.format(logreg.score(X_train, y_train)))
print('Accuracy of Logistic regression classifier on test set: {:.2f}'.format(logreg.score(X_test, y_test)))
```

6. Decision Tree

Decision Tree Classifier adalah algoritma klasifikasi yang dapat digambarkan dengan skema pohon. Klasifikasi ini akan membagi data training menjadi node hingga leaf-node. Data yang sudah melalui partisi node nantinya akan berakhir ke leaf-node, yang kemudian akan mendominasi leaf-node dengan 1 kategori data.

```python
from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
print('Accuracy of Decision Tree classifier on training set: {:.2f}'.format(clf.score(X_train, y_train)))
print('Accuracy of Decision Tree classifier on test set: {:.2f}'.format(clf.score(X_test, y_test)))
```

7. K Nearest Neighbor

   KNeighbors Classifier secara sederhana adalah algoritma yang melakukan klasifikasi berdasarkan voting kemiripan dari kategori lainnya. Data yang masuk kedalam algoritma ini akan diberikan nilai K dan ditempatkan pada area data yang memiliki kemiripan nilai K. Nilai K dapat diubah sesuai keinginan, contoh jika K=1 maka data akan ditempatkan pada 1 tetangga yang nilainya paling mendekati.

```python
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'.format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'.format(knn.score(X_test, y_test)))
```

8. Naïve Bayes

   Algoritma Naive Bayes adalah algoritma sederhana dimana algoritma ini menghitung probabilitas terjadinya skenario B jika skenario A sudah berjalan, sehingga dalam klasifikasi, algoritma ini diasumsikan akan melakukan klasifikasi secara merata.

   Gaussian Naive-Bayes adalah klasifikasi Naive-Bayes yang digunakan jika kategorinya mengikuti penyebaran Gaussian. Algoritma lainnya adalah BernoulliNB dan MultinomialNB

```python
from sklearn.naive_bayes import GaussianNB

gnb = GaussianNB()
gnb.fit(X_train, y_train)
print('Accuracy of GNB classifier on training set: {:.2f}'.format(gnb.score(X_train, y_train)))
print('Accuracy of GNB classifier on test set: {:.2f}'.format(gnb.score(X_test, y_test)))
```

Challenge

9. Cobalah hilangkan data dengan kualitas 3,4,7, dan 8 sehingga data hanya memiliki 2 output kategori, dan bagaimana hasil prediksinya? Manakah prediksi data yang lebih baik? (Boleh pilih gunakan salah satu algoritma).
10. Menampilkan Confusion Matrix

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
```

Menggunakan salah satu algoritma, yaitu Naïve Bayes.

```
y_pred = gnb.predict(X_test)

mat = confusion_matrix(y_test, y_pred)
sns.heatmap(mat.T, square = True, annot = True, cmap = 'Blues')
plt.xlabel('true label')
plt.ylabel('predicted label')
```

11. Bandingkan juga dengan data white wine yang menggunakan multiple output dan binary output!

Unsupervised Learning

Perbedaan Unsupervised Learning dengan Supervised Learning adalah data yang diberikan tidak memliki label sehingga tidak ada bantuan untuk sistem dalam melakukan klasifikasi. Biasanya dilakukan untuk melakukan clustering data.

12. Contoh melakukan clustering acidity pada data wine tanpa label quality.

```
winecluster = dataset.copy()
winecluster = winecluster.drop(columns='quality')
winecluster
```

13. Untuk mengetahui seberapa besar penyebaran data untuk clustering dapat dilihat melalui grafik.

```
import matplotlib.pyplot as plt

plt.scatter(winecluster['volatile acidity'],winecluster['fixed acidity'])
plt.xlim(0,2)
plt.ylim(0,20)
plt.show()
```

14. Selanjutnya mengambil kolom yang akan dilakukan clustering, dan mengimport library KMeans dari SKlearn.

```
x = dataset.iloc[:,0:2]
x.info()

from sklearn.cluster import KMeans
```

Sebelum melakukan clustering, tentukan terlebih dahulu berapa banyak cluster data yang akan dilakukan. Cara untuk menentukan angka clustering dapat dengan beberapa cara, diantaranya adalah dengan elbow method.

Elbow method ini adalah teknik mencari K cluster dengan membuat grafik berdasarkan SSE (Sum of Squared Error) atau WCSS. SSE adalah penjumlahan kuadrat jarak antara cluster dengan titik tengah dari semua jarak cluster. Kemudian SSE ini akan digambarkan menggunakan grafik, dan cari titik dimana grafik tersebut turun paling jauh, dan membentuk siku.

15. Mencari jumlah K dengan Elbow Method.

```
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(x)
    wcss.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss)
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

16. Pada grafik tersebut, titik yang membentuk siku adalah cluster 4. Cobalah membuat KMeans dengan menggunakan K = 4. Untuk melakukan clustering, function yang digunakan adalah fit_predict, berbeda dengan supervised learning yang menggunakan fit_transform.

```
kmeans = KMeans(4)
kmeans.fit(x)

identified_clusters = kmeans.fit_predict(x)
identified_clusters
```

17. Tampilkan hasil pada grafik untuk melihat hasil clustering yang dilakukan oleh KMeans Clustering.

```
wine_clusters = winecluster.copy()
wine_clusters['Clusters'] = identified_clusters
plt.scatter(winecluster['fixed acidity'],winecluster['volatile acidity'],c=wine_clusters['Clusters'],cmap='Spectral')
```

Challenge

18. Coba lakukan clustering berdasarkan pada kolom free sulfur dioxide dan total sulfur dioxide, dan tampilkan hasilnya. Apakah nilai K-nya sama dengan clustering sebelumnya?
19. Cobalah lakukan clustering pada data whitewine dan bandingkan hasilnya.

PENGUMPULAN

1. File yang dikumpulkan terdiri dari:
    a. File project (.ipynb)
    b. File PDF berisi screenshot output dan jawaban (jika ada pertanyaan)
2. File di-compress (.zip) dan diberi nama **KODEMATAKULIAH_KELAS_NIM_NAMA_WEEK KE-XX.zip** (contoh: IS5411_A_13110310017_Monika Evelin Johan_Week-01.zip).

REFERENSI

Deitel, P., & Deitel, H. (2020). Intro to Python for Computer Science and Data Science. Pearson Education.