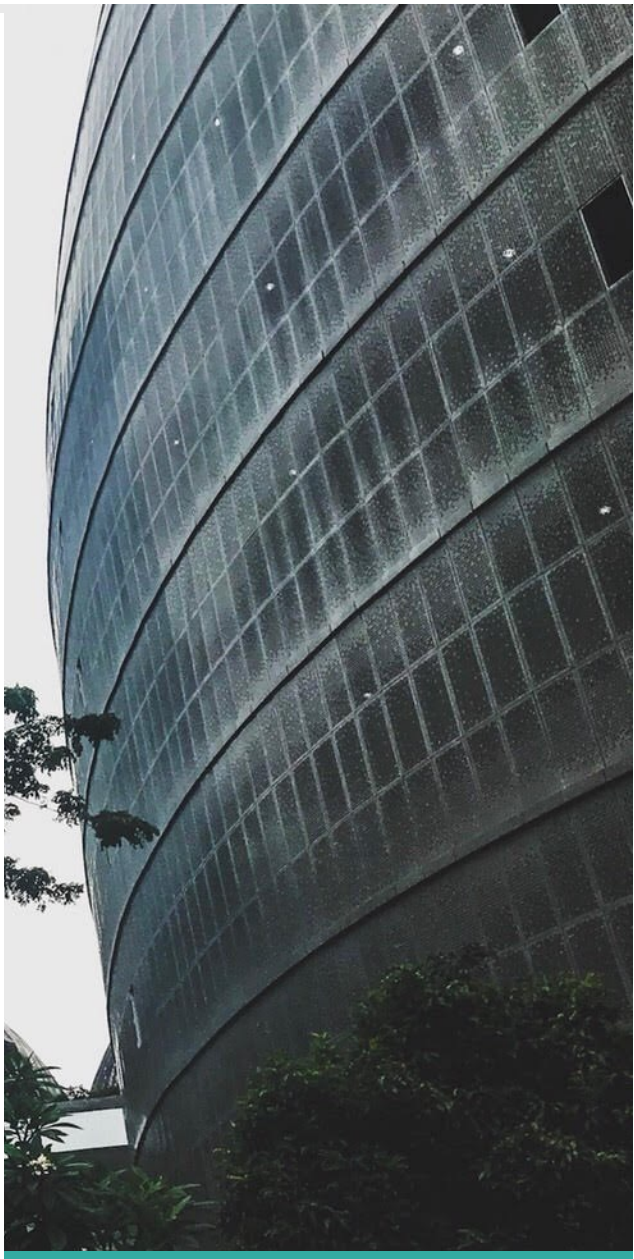
MODUL PRAKTIKUM

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 9
CLASSIFICATION

DESKRIPSI TEMA

1. Decision Tree Algorithm
2. Random Forest Algorithm

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Students are able to illustrate data modeling problems using classification techniques (C4)

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 untuk mengimport library Numpy, Pandas, Matplotlib, dan Seaborn
2. Import dataset menggunakan file “titanic_train.csv”.

```
1 titanic = pd.read_csv('titanic_train.csv', delimiter=";")
2 titanic
```

Note: gunakan delimiter sesuai settingan pada komputer Anda (bisa “;” atau “,”).

Data Exploration

3. Melihat deskripsi data

```
1 titanic.info()
```

Perhatikan bahwa terdapat beberapa atribut yang memiliki data null. Untuk mendapatkan hasil yang baik kita perlu mencoba memperbaiki data tersebut.

4. Mengisi Data Null

```
#Kolom Age
#mengisi data null pada kolom Age dengan rata - rata usia

rata_umur = titanic['Age'].mean()
titanic['Age'] = titanic['Age'].fillna(rata_umur)

#memeriksa apakah masih ada yang null pada kolom Age
titanic['Age'].isna().sum()
```

```
#Kolom Embarked
#mengisi data null pada kolom Embarked dengan nilai yang paling banyak muncul (modus)

modus = titanic['Embarked'].mode()[0]
titanic['Embarked'] = titanic['Embarked'].fillna(modus)
#memeriksa apakah masih ada yang null pada kolom Embarked
titanic['Embarked'].isna().sum()
```

5. Periksa apakah semua data sudah tidak ada yang null.

```
titanic.info()
```

Selain data null, terdapat beberapa atribut yang tidak akan digunakan sehingga perlu dihapus. Di samping itu terdapat kolom “Sex” dan kolom “Embarked” yang perlu diubah formatnya menjadi integer agar dapat digunakan pada modeling.

6. Data Cleansing

```
#Menghapus kolom Cabin

titanic.drop('Cabin', axis=1, inplace=True)
#axis = 1 menandakan kolom, axis = 0 menandakan baris
```

```
#membuang kolom Ticked
titanic.drop('Ticket', axis=1, inplace=True)
```

```
#menghapus kolom yang tidak digunakan
titanic.drop('Name', axis=1, inplace=True)
```

7. Konversi data

```
#mengkonver data kolom Sex, male = 0; female = 1

titanic['Sex'] = titanic['Sex'].replace(['male', 'female'], [0, 1])
titanic.head()
```

```
#mengkonver data kolom Embarked, S = 1, C = 2, Q = 3

titanic['Embarked'] = titanic['Embarked'].replace(['S', 'C', 'Q'], [1, 2, 3])
titanic.head()
```

```
titanic['Fare'] = titanic['Fare'].str.replace(".","")
```

```
titanic.head()
```

Periksa apakah semua data sudah siap untuk digunakan.

Mencari Korelasi Data

8. Kita bisa melakukan analisis korelasi untuk mengetahui korelasi antar data sehingga bisa memilih hanya atribut yang paling berkorelasi dengan data target.

```
#untuk mencari tahu variabel / feature yang paling mempengaruhi target (Survived)

f, ax = plt.subplots(figsize=(10,10))

sns.heatmap(titanic.astype(float).corr(), linewidth=0.25, vmax=1.0, square=True, linecolor='black', annot=True)
```

Analisislah hasil dari Pearson Correlation tersebut. Mana saja atribut / feature yang menurutmu paling memiliki korelasi pada data target “Survived”?

Membangun Model

9. Menentukan nilai X dan Y

```
#membagi data X dan Y (x = variabel independen; y = variabel dependen/target)

x = titanic[['Sex', 'Parch', 'Fare', 'Embarked']]
#x hanya menggunakan data yang memiliki korelasi tinggi dengan Survived
y = titanic['Survived']
```

10. Membagi data training dan data testing

```
#membagi data training dan testing

from sklearn.model_selection import train_test_split

train_x, test_x, train_y, test_y = train_test_split(x, y, random_state = 0)
print(train_x.shape)
print(test_x.shape)
print(train_y.shape)
print(test_y.shape)
```

11. Modeling dengan algoritma Decision Tree

```
#Modeling dengan Decision Tree

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()
dt.fit(train_x, train_y)

#mengevaluasi hasil prediksi
y_prediksi3 = dt.predict(test_x)
akurasi_dt = dt.score(test_x, test_y)
akurasi_dt
```

12. Menampilkan struktur Decision Tree

```
#menampilkan struktur Decision Tree

from sklearn import tree

feature_names = x.columns[:4]
target_names = y.unique().tolist()

target_names = [str(x) for x in target_names]

fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(dt,
                   feature_names = feature_names,
                   class_names = target_names,
                   filled = True)
```

13. Modeling dengan algoritma Random Forest

```
#modeling dengan Random Forest

from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators = 10, random_state = 42)
rf.fit(train_x, train_y)

#mengevaluasi hasil prediksi
y_prediksi4 = rf.predict(test_x)
akurasi_rf = rf.score(test_x, test_y)
akurasi_rf
```

Implementasi Machine Learning

Menggunakan dataset **titanic_test.csv** yang belum ada labelnya (tidak ada kolom Survived).

14. Lakukan langkah yang sama mulai dari import data, data cleansing, sampai menentukan data X. Di sini kita tidak menentukan data Y karena tidak ada kolom target, yang akan dilakukan adalah prediksi.

```
titanic_test = pd.read_csv('titanic_test.csv', delimiter=';')
titanic_test.head()
```

```
titanic_test.info()
```

- Periksa data apa saja yang memiliki Null. Lakukan langkah yang sama seperti di atas untuk mengisi data tersebut. Untuk kolom Fare bisa mengisi data Null dengan nilai 0.
- Jangan lupa untuk mengubah format data Sex dan Embarked.

15. Menentukan data X.

```
#menentukan variabel X dan Y

x1 = titanic_test[['Sex','Parch', 'Fare', 'Embarked']]
```

16. Implementasi menggunakan Random Forest untuk prediksi

```
#implementasi Random Forest pada dataset untuk melakukan prediksi

titanic_prediction = rf.predict(x1)
titanic_prediction
```

17. Buat ke dataframe

```
#buat ke dataframe
df_titanic_prediction = pd.DataFrame({"Predicted Survived" : titanic_prediction})
df_titanic_prediction
```

18. Menggabungkan dataset dengan hasil prediksi

```
#menggabungkan hasil prediksi dengan dataset

result = pd.concat([titanic_test, df_titanic_prediction], ignore_index = False, axis = 1)
result
```

19. Menyimpan hasil yang baru pada file CSV

```
#menyimpan hasil yang baru ke dalam file csv
export_result2 = result.to_csv('Hasil Prediksi Titanic CSV.csv')
print(export_result2)
```

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.