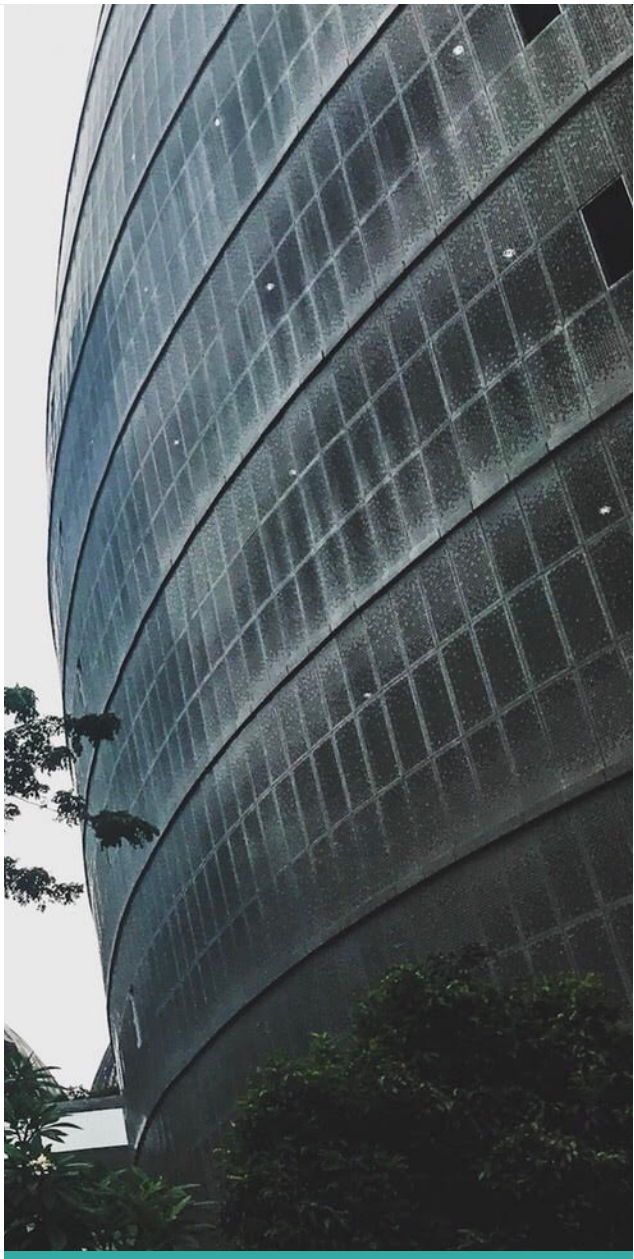
MODUL PRAKTIKUM

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 4
DM METHODOLOGY

DESKRIPSI TEMA

1. Data modeling algorithms implementation in certain case study

CAPAIAN PEMBELAJARAN MINGGUAN (SUB-CAPAIAN PEMBELAJARAN)

Students are able to examine Data Modeling Methodology (C_3)

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.

Linear Regression

Linear Regression merupakan salah satu algoritma yang digunakan pada data modelling. Algoritma ini sangat sederhana karena hanya memiliki 1 independent variable (variable bebas). Algoritma sederhana ini biasanya digunakan untuk melakukan prediksi seperti prediksi suhu, prediksi harga, dsb.

Rumus yang digunakan untuk algoritma ini dapat dituliskan sebagai berikut **y = mx + a + b** , dimana y adalah dependent variable, m adalah slope (tingkat kemiringan) dari garis, x adalah independent variable, a adalah intercept, dan b adalah error.

Berikut adalah contoh membuat Linear Regression sederhana menggunakan numpy dan matplotlib.

2. Import library Matplotlib

```
1 import matplotlib.pyplot as plt
```

3. Menentukan variable independent dan dependen

```
#independent_var merupakan variabel yang memperngaruhi dependent_var
#independent_var sebagai data x, dependent_var sebagai data y

independent_var = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
dependent_var = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])
```

```
# number of observations/points
n = np.size(independent_var)
```

4. Menghitung cross deviation dan squared deviation

```
#menghitung rata - rata (mean) untuk menghitung cross deviation dan squared deviation

# mean of x and y vector
mean_indep_var = np.mean(independent_var)
mean_dep_var = np.mean(dependent_var)

#Fungsi cross deviation dan squared deviation adalah untuk membuat garis linear yang menggambarkan posisi
#dependent variable yang akan diprediksi

# calculating cross-deviation and deviation about x
cross_deviation = np.sum(dependent_var*independent_var) - n*mean_dep_var*mean_indep_var
squared_deviation = np.sum(independent_var*independent_var) - n*mean_indep_var*mean_indep_var
```

5. Menampilkan penyebaran data independent menggunakan scatter dari Matplotlib.

```
# calculating regression coefficients
b_1 = cross_deviation / squared_deviation
b_0 = mean_dep_var - b_1*mean_indep_var

# plotting the actual points as scatter plot
plt.scatter(independent_var, dependent_var, color = "m", marker = "o", s = 30)

# predicted response vector
y_pred = b_0 + b_1*independent_var
```

```
# plotting the regression line
plt.plot(independent_var, y_pred, color = "g")

# putting labels
plt.xlabel('x')
plt.ylabel('y')

# function to show plot
plt.scatter(independent_var, dependent_var, color = "m", marker = "o", s = 30)
plt.show()
```

Setelah mengetahui konsep dari Linear Regression, berikutnya adalah menerapkan konsep linear regression untuk membuat model data yang akan digunakan dalam memprediksi tempatur suhu. Data

yang digunakan sebagai independent variable adalah MinTemp (x) sedangkan dependent variable yang adalah MaxTemp (y).

6. Prediksi suhu ini akan menggunakan library dari scikit-learn untuk mempermudah penerapan linear regression.

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

7. Membaca Data

```
dataset = pd.read_csv("weather.csv", header=0)
dataset.info()
dataset
```

8. Menentukan variable x dan y.

```
#Kita akan melakukan prediksi MaxTemp berdasarkan MinTemp
#maka MaxTemp sebagai dependent variable (y) dan MinTemp sebagai independent variable (x)
#kolom yang akan digunakan adalah MinTemp dan MaxTemp

dataTemp = dataset[['MinTemp','MaxTemp']]
dataTemp.info()
```

9. Melihat korelasi data menggunakan Scatter Plot

```
#melihat korelasi data dengan Scatter Plot

plt.scatter(dataTemp['MinTemp'], dataTemp['MaxTemp'])
plt.xlabel('MinTemp')
plt.ylabel('MaxTemp')
plt.title('Scatter Plot MinTemp vs MaxTemp')
plt.show()
```

Pada scatter plot tersebut dapat dilihat bahwa hubungan dari mintemp dan max temp memiliki korelasi positif yang signifikan. Dari gambar tersebut terlihat bahwa semakin besar minimum temperatur pada suhu, maka maksimum temperatur juga cenderung meningkat.

10. Tahap berikutnya adalah data modelling menggunakan Linear Regression.

```
x = dataTemp['MinTemp'].values.reshape(-1,1)
y = dataTemp['MaxTemp'].values.reshape(-1,1)

#reshape(-1,1) untuk mengubah susunan data menjadi row ke samping.
```

```
#membagi data training dan data testing

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4)
```

```
LinReg = LinearRegression()
LinReg.fit(x_train,y_train)
LinReg.score(x_test,y_test)
```

Catatlah, berapa hasil akurasi yang Anda peroleh? (Masing – masing bisa mendapatkan hasil yang berbeda).

11. Function score pada object LinReg tersebut merupakan tingkat akurasi dari data modelling menggunakan Linear Regression untuk melakukan prediksi terhadap suhu maksimal dari suhu minimal.
    Jika diimplementasikan, maka output dari Linear Regression ini akan berbentuk seperti berikut.

```
y_predict = LinReg.predict(x_test)
plt.scatter(x_test, y_test)
plt.plot(x_test, y_predict, c='r')
plt.xlabel('MinTemp')
plt.ylabel('MaxTemp')
plt.title('Plot MinTemp vs MaxTemp')
```

Logistic Regression

Logistic Regression merupakan algoritma lain yang dapat digunakan dalam data modelling, dimana outputnya berupa klasifikasi yang dihasilkan dari probabilitas.

Pada Linear Regression, data modellingnya hanya menggunakan 1 variable independent, tetapi Logistic Regression menggunakan lebih dari 1 variable independent.

Menggunakan contoh data temperatur yang sama, Logistic Regression dapat melakukan prediksi akan terjadinya hujan atau tidak berdasarkan variable-variable yang ada pada dataset tersebut.

12. Copy dataset

```
dataLogistic = dataset.copy()
dataLogistic.info()
```

Analisislah data tersebut. Perhatikan bahwa terdapat beberapa kolom yang memiliki nilai null atau missing value. Maka kita perlu memperbaiki data tersebut dengan mengisi nilai 0 pada data yang null.

13. Mengatasi data null

```
dataLogistic['WindGustSpeed'] = dataLogistic['WindGustSpeed'].fillna(0)
dataLogistic['WindSpeed9am'] = dataLogistic['WindSpeed9am'].fillna(0)
dataLogistic['Sunshine'] = dataLogistic['Sunshine'].fillna(0)
dataLogistic.info()
dataLogistic
```

14. Melakukan encoding

Encoding adalah proses memberikan kode atau seperti inisial untuk value yang ada pada kolom, sehingga valuenya menjadi numerical. Langkah ini sangat penting dalam Logistic Regression karena model ini hanya menerima inputan dalam tipe numerical. Dalam contoh ini, encoding bertujuan untuk mengganti value text ke dalam kode yang akan ditentukan otomatis oleh sistem, misalnya pada kolom WindGustDir, untuk data 'NW' akan diberikan label 1, ENE label 2 dan seterusnya.

```
def encode_data(feature_name):
    '''

    This function takes feature name as a parameter and returns mapping dictionary to replace(or map) categorical data with

    '''

    mapping_dict = {}

    unique_values = list(dataLogistic[feature_name].unique())

    for idx in range(len(unique_values)):

        mapping_dict[unique_values[idx]] = idx

    return mapping_dict


dataLogistic['RainToday'].replace({'No':0, 'Yes': 1}, inplace = True)

dataLogistic['RainTomorrow'].replace({'No':0, 'Yes': 1}, inplace = True)

dataLogistic['WindGustDir'].replace(encode_data('WindGustDir'),inplace = True)

dataLogistic['WindDir9am'].replace(encode_data('WindDir9am'),inplace = True)

dataLogistic['WindDir3pm'].replace(encode_data('WindDir3pm'),inplace = True)
```

15. Memeriksa data. Perhatikan bahwa isi data telah berubah.

```
dataLogistic
dataLogistic.info()
```

16. Correlation Analysis

Analisa korelasi bertujuan untuk menemukan hubungan antara data pada kolom, nantinya analisa ini dapat membantu untuk pemilihan kolom yang akan digunakan kedalam model. Pada contoh ini , semua kolom akan tetap dipakai.

```
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
plt.figure(figsize=(20,20))
sns.heatmap(dataLogistic.corr(), linewidths=0.5, annot=False, fmt=".2f", cmap = 'viridis')

#linewidths untuk tebal garis; linecolor untuk garis diluar visualisasi; annot untuk label tiap baris;
#cmap untuk pilih tema warna
#lihat pilihan warna di seaborn: https://seaborn.pydata.org/tutorial/color_palettes.html
```

17. Menentukan nilai x dan y

    Karena model yang dibuat akan memprediksi turunnya hujan, maka pisahkan kolom 'RainTomorrow' sebagai dependent variable. Setelah menentukan dependent (y) dan independent (x) variable, bagi data tersebut kedalam 4 variable untuk data train dan data test. Pada contoh ini jumlah data yang digunakan untuk training adalah 60% dari total data dan model akan melakukan prediksi untuk 40% dari data.

```
X = dataLogistic.drop(['RainTomorrow'],axis=1)
y = dataLogistic['RainTomorrow']

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.4, random_state = 0)

print("Length of Training Data: {}".format(len(X_train)))
print("Length of Testing Data: {}".format(len(X_test)))
```

18. Memasukkan data kedalam model Logistic Regression.

```
logreg = LogisticRegression(solver='liblinear')
logreg.fit(X_train, y_train)
```

Karena data berjumlah sedikit maka hasil akurasi yang dihasilkan juga sangat tinggi. Untuk itu cobalah menggunakan dataset lain dengan jumlah data yang lebih besar untuk melihat hasil Logistic Regression dalam jumlah data yang besar.

19. Evaluasi data

Tahap terakhir adalah melakukan review terhadap model yang dibuat. Tahap ini dapat menggunakan akurasi dari model atau ROC Curve.

Berikut ini adalah 4 kemungkinan yang dapat terjadi dalam prediksi yang dilakukan Logistic Regression, yaitu:

a. True Positive: prediksi hujan, dan turun hujan

b. True Negative: prediksi tidak hujan, dan tidak turun hujan

```
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

```
print("Accuracy : ", logreg.score(X_test,y_test))

from sklearn.metrics import roc_curve

y_pred_logreg_proba = logreg.predict_proba(X_test)[:,1]
fpr, tpr, thresholds = roc_curve(y_test, y_pred_logreg_proba)
plt.figure(figsize=(6,4))
plt.plot(fpr,tpr,'-g',linewidth=1)
plt.plot([0,1], [0,1], 'k--' )
plt.title('ROC curve for Logistic Regression Model')
plt.xlabel("False Positive Rate")
plt.ylabel('True Positive Rate')
plt.show()
```

- **Akurasi** adalah jumlah kebenaran dari prediksi yang dihasilkan, kemudian dibagi oleh total semua prediksi baik yang benar maupun yang salah.
- **ROC Curve** adalah grafik yang memperlihatkan tingkat prediksi yang benar dan yang salah terhadap batas yang ditentukan. grafik ini sudah disediakan oleh library sklearn dan cukup diimport kedalam project python.
- **y_pred_logreg_proba** adalah variable yang mengeluarkan skor probabilitas per line data.
- **threshold** yang dihasilkan nantinya dapat digunakan untuk melakukan klasifikasi data agar model berikutnya dapat memprediksi lebih akurat.

  TUGAS :

  Dengan menggunakan dataset lain yaitu : harga_mobil.csv lakukan prediksi harga dengan menggunakan metode Linier Regression dan Logistic Regression

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.