TUGAS BESAR TAHAP 2 CLASSIFICATION

Laporan

Dibuat untuk memenuhi tugas mata kuliah Pembelajaran Mesin

oleh :

Farhan Anas	(1301183427)
Gilang Ramadhan	(1301184376)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2021

1. Formulasi Masalah

   Permasalahan yang ingin diselesaikan pada tugas besar tahap 2 ini adalah bagaimana cara mengklasifikasikan data dengan memprediksi kelas dari masing-masing data seakurat mungkin dimana dalam pengujian kali ini terdapat 2 kelas yakni tidak bersalju besok (direpresentasikan dengan nilai 0) dan bersalju besok (direpresentasikan dengan nilai 1).

2. Eksplorasi dan Persiapan Data
   Eksplorasi dan Persiapan data adalah proses pre-processing terhadap data agar dataset menghasilkan output yang lebih baik. Proses ini dilakukan terhadap 2 dataset yaitu dataset train dan dataset test. Pre-processing yang digunakan antara lain :

   A. Drop Kolom yang Tidak Dibutuhkan

      Kolom yang tidak dipakai akan didrop sehingga dapat memudahkan proses pemodelan. Kolom yang tidak dipakai adalah kolom yang berisi data yang tidak dapat diubah ke bentuk numerik karena pemodelan hanya dapat memproses data yang bersifat numerik.

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	KecepatanAnginTerkencang	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am	Tekanan3pm	Awan9am	Awan3pm	S
0	10.4	15.5	4.8	NaN	NaN	24.0	0.0	13.0	78.0	76.0	1020.1	1018.5	NaN	NaN	
1	9.0	17.0	8.0	2.6	7.4	NaN	13.0	20.0	80.0	61.0	1015.2	1014.6	7.0	5.0	
2	18.2	32.0	0.0	NaN	NaN	44.0	15.0	26.0	62.0	42.0	NaN	NaN	NaN	NaN	
3	7.3	24.5	0.0	8.4	10.4	54.0	13.0	19.0	25.0	17.0	1019.2	1016.9	1.0	7.0	
4	5.9	20.3	0.0	3.6	12.6	37.0	22.0	19.0	55.0	48.0	1019.7	1014.7	2.0	6.0	

   B. Drop Data yang Memiliki Nilai sama

      Data yang memiliki nilai/value sama akan dihapus karena data tersebut tidak akan berdampak ke hasil pemodelan sehingga akan lebih baik jika dihapus untuk mempercepat proses pemodelan.

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	KecepatanAnginTerkencang	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am	Tekanan3pm	Awan9am	Awan3pm	Suhu9
3	7.3	24.5	0.0	8.4	10.4	54.0	13.0	19.0	25.0	17.0	1019.2	1016.9	1.0	7.0	15
4	5.9	20.3	0.0	3.6	12.6	37.0	22.0	19.0	55.0	48.0	1019.7	1014.7	2.0	6.0	12
5	14.4	21.8	0.0	3.2	4.4	39.0	19.0	20.0	63.0	52.0	1016.1	1012.5	7.0	7.0	16
6	7.7	18.7	0.2	5.6	9.7	46.0	19.0	28.0	69.0	31.0	1011.3	1008.8	1.0	1.0	11
8	18.4	35.3	0.0	10.0	12.5	33.0	11.0	13.0	44.0	18.0	1017.9	1013.4	0.0	0.0	23

   C. Drop Data yang Memiliki Nilai Null

      Data yang memiliki nilai null akan dihapus karena data tersebut dapat membuat hasil pemodelan menjadi kurang relevan yang disebabkan nilai dari data null tidak memiliki nilai yang pasti.

	SuhuMin	SuhuMax	Hujan	Penguapan	SinarMatahari	KecepatanAnginTerkencang	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am	Tekanan3pm	Awan9am	Awan3pm	Suhu9
3	7.3	24.5	0.0	8.4	10.4	54.0	13.0	19.0	25.0	17.0	1019.2	1016.9	1.0	7.0	15
4	5.9	20.3	0.0	3.6	12.6	37.0	22.0	19.0	55.0	48.0	1019.7	1014.7	2.0	6.0	12
5	14.4	21.8	0.0	3.2	4.4	39.0	19.0	20.0	63.0	52.0	1016.1	1012.5	7.0	7.0	16
6	7.7	18.7	0.2	5.6	9.7	46.0	19.0	28.0	69.0	31.0	1011.3	1008.8	1.0	1.0	11
8	18.4	35.3	0.0	10.0	12.5	33.0	11.0	13.0	44.0	18.0	1017.9	1013.4	0.0	0.0	23

D. Modifikasi Data ke Bentuk Numerik

Mengubah kolom yang tidak bernilai numerik menjadi data yang bernilai numerik. Langkah ini dilakukan karena pemodelan hanya dapat memproses data yang bersifat numerik.

arMatahari	KecepatanAnginTerkencang	KecepatanAngin9am	KecepatanAngin3pm	Kelembaban9am	Kelembaban3pm	Tekanan9am	Tekanan3pm	Awan9am	Awan3pm	Suhu9am	Suhu3pm	BersaljuHariIni	BersaljuBesok
10.4	54.0	13.0	19.0	25.0	17.0	1019.2	1016.9	1.0	7.0	15.3	23.2	0	0
12.6	37.0	22.0	19.0	55.0	48.0	1019.7	1014.7	2.0	6.0	12.4	18.1	0	0
4.4	39.0	19.0	20.0	63.0	52.0	1016.1	1012.5	7.0	7.0	16.7	21.1	0	0
9.7	46.0	19.0	28.0	69.0	31.0	1011.3	1008.8	1.0	1.0	11.3	18.3	0	0
12.5	33.0	11.0	13.0	44.0	18.0	1017.9	1013.4	0.0	0.0	23.7	34.9	0	0

E. Hapus Outlier

Data outlier adalah data yang memiliki persebaran sangat jauh dari rata-rata data yang ada namun jumlahnya hanya sedikit. Data-data tersebut akan dihapus agar hasil pemodelan menjadi lebih baik.

```
#Menampilkan jumlah data beserta dengan outliers
print("jumlah data dengan outliers:", df_salju_train.shape)
#Drop outliers
df_salju_train = df_salju_train[(np.abs(stats.zscore(df_salju_train)) < 3).all(axis=1)]
#Menampilkan jumlah data setelah drop outliers
print("jumlah data tanpa outliers:", df_salju_train.shape)
```

```
jumlah data dengan outliers: (43677, 18)
jumlah data tanpa outliers: (41348, 18)
```

F. Scalling

Semua nilai dari data akan diubah menjadi data yang memiliki range nilai 0-1 sehingga dapat memudahkan proses pemodelan.

```
[ ] #Mengubah nilai data pada tiap kolom menjadi range 0-1
    df_salju_train_MinMax = (df_salju_train - df_salju_train.min()) / (df_salju_train.max() - df_salju_train.min())
    #Menyimpan kolom-kolom yang bukan target kedalam variabel x_train
    x_train = df_salju_train_MinMax.iloc[:, :-1].values
    #Menyimpan kolom yang merupakan target ("BersaljuBesok") kedalam variabel y_train
    y_train = df_salju_train_MinMax.iloc[:, -1].values
    #Menampilkan data
    x_train
```

```
array([[0.36901408, 0.47519582, 0.        , ..., 0.41666667, 0.48837209,
        0.        ],
       [0.32957746, 0.36553525, 0.        , ..., 0.34114583, 0.35658915,
        0.        ],
       [0.56901408, 0.40469974, 0.        , ..., 0.453125  , 0.43410853,
        0.        ],
       ...,
       [0.56619718, 0.51958225, 0.        , ..., 0.56770833, 0.54780362,
        0.        ],
       [0.72957746, 0.45430809, 0.        , ..., 0.59114583, 0.45994832,
        0.        ],
       [0.46760563, 0.61357702, 0.        , ..., 0.58333333, 0.64341085,
        0.        ]])
```

3. Pemodelan

Terdapat 4 proses yang dilakukan pada tahap pemodelan, yakni pertama-tama menentukan model yang akan digunakan untuk mengklasifikasikan data yang dalam hal ini menggunakan metode decision tree classifier karena cenderung simpel untuk digunakan dibandingkan dengan model lainnya. Selanjutnya melakukan data train dengan menggunakan dataset salju_train yang sudah diolah dan menggunakan metode fit agar dataset sesuai dengan model yang digunakan.

```
#Membuat sebuah objek menggunakan metode decision tree classifier
model = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
#Melakukan data train dengan dataset yang ada dengan menggunakan metode fit
model.fit(x_train, y_train)
```

Lalu selanjutnya mengklasifikasi data test dengan menggunakan dataset salju_test yang sudah diolah untuk menghasilkan tabel berisi hasil klasifikasi berdasarkan model yang digunakan. Terakhir, menggabungkan tabel data prediksi dengan data yang sebenarnya untuk dapat melihat perbandingan hasil klasifikasinya.

```
#Melakukan klasifikasi pada data test
y_pred = model.predict(x_test)
#Menggabungkan data test dengan hasil prediksi untuk melakukan perbandingan
df_hasil = pd.concat([pd.DataFrame(y_test, columns=['test']) ,pd.DataFrame(y_pred, columns=['predict'])], axis=1)
```

4. Evaluasi

Pada tahap evaluasi ini digunakan confusion matrix yang bertujuan untuk mencari nilai akurasi, recall, precision dan juga f1 score. Nantinya, nilai-nilai ini akan digunakan sebagai acuan untuk mengetahui seberapa baik sebuah model jika dibandingkan dengan model-model lainnya.

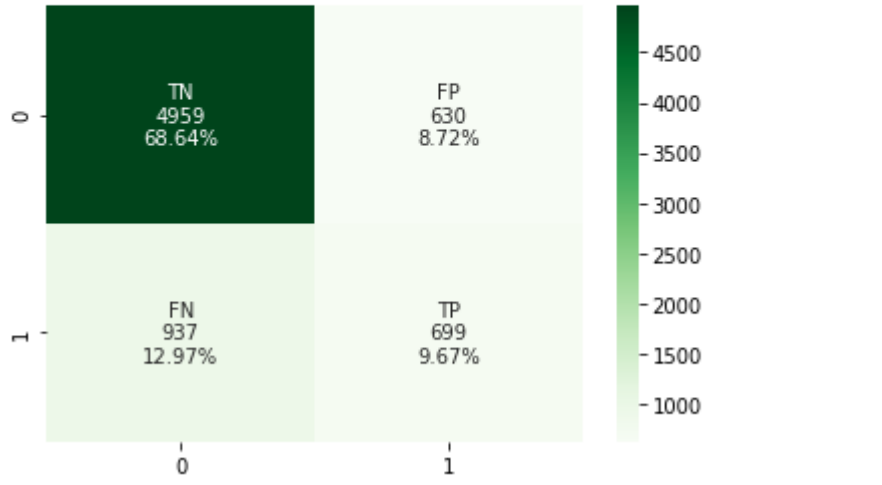
```
#Membuat confusion matrix
cm = confusion_matrix(y_test, y_pred)

#Menghitung accuracy dengan rumus accuracy = (TP + TN ) / (TP + FP + FN + TN)
accuracy = accuracy_score(y_test, y_pred) * 100
print('Accuracy: %0.2f' % accuracy , '%')
#Menghitung precision dengan rumus precision = (TP) / (TP + FP)
precision = precision_score(y_test, y_pred) * 100
print('Precision: %0.2f' % precision , '%')
#Menghitung recall dengan rumus recall = (TP) / (TP + FN)
recall = recall_score(y_test, y_pred ) *100
print('Recall: %0.2f' % recall , '%')
#Menghitung f1 score dengan rumus f1 score = 2 * (Recall * Precission) / (Recall + Precission)
f1 = f1_score(y_test, y_pred , '%') *100
print('F1 score: %0.2f' % f1 , '%')
```

5. Eksperimen

Eksperimen yang dilakukan ada 3 macam pemodelan. Yaitu : Decision Tree Learning, Naive Bayes, Logistic Regression. Dalam melakukan eksperimen, semua pre-processing data disamakan agar dapat mengetahui pemodelan mana yang paling cocok untuk dataset yang dipakai. Maka diperoleh hasil sebagai berikut:

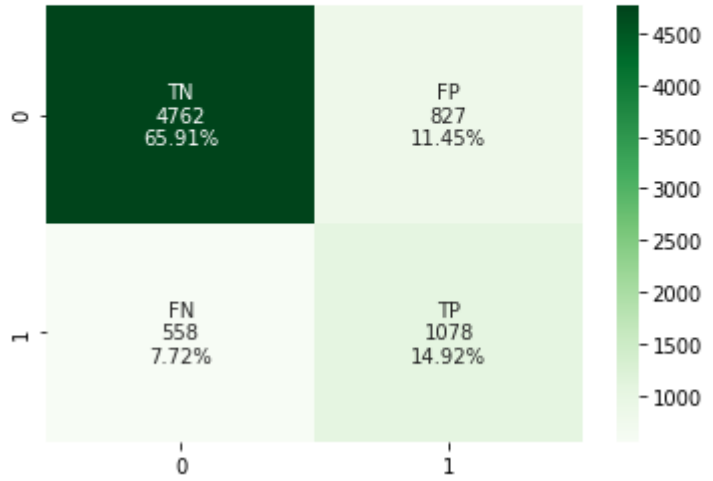
A. Decision Tree Learning

```
Accuracy: 78.31 %
Precision: 52.60 %
Recall: 42.73 %
F1 score: 47.15 %
<matplotlib.axes._subplots.AxesSubplot at 0x7fde8f6e4bd0>
```



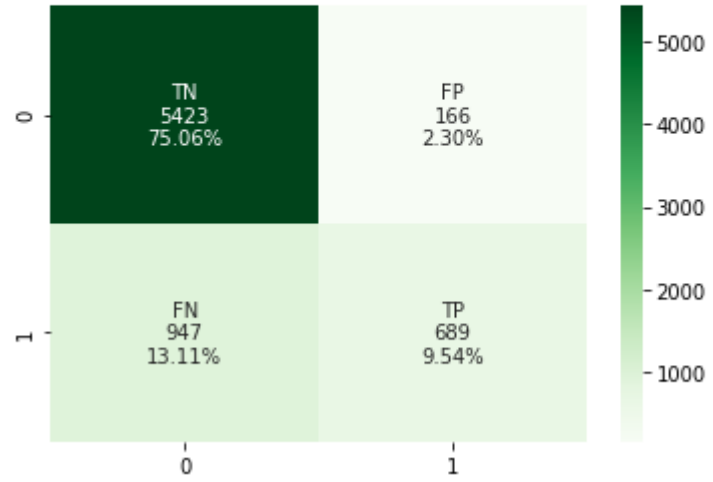
B. Naive Bayes

```
Accuracy: 80.83 %
Precision: 56.59 %
Recall: 65.89 %
F1 score: 60.89 %
<matplotlib.axes._subplots.AxesSubplot at 0x7fde8f6cdc10>
```



C. Logistic Regression

```
Accuracy: 84.60 %
Precision: 80.58 %
Recall: 42.11 %
F1 score: 55.32 %
<matplotlib.axes._subplots.AxesSubplot at 0x7fde8f66e150>
```



Dalam studi kasus penentuan turun salju, nilai evaluasi yang dilihat adalah nilai Accuracy. Ini dikarenakan prediksi turun salju hanya memfokuskan berapa banyak hasil prediksi yang sesuai. Berbeda dengan contoh kasus lain seperti “Pengecekan Pasien Positif Corona” yang harus mempertimbangkan nilai dari F1-Score.

6. Kesimpulan

Kesimpulan yang didapat adalah pemodelan terbaik untuk prediksi turun salju adalah pemodelan yang memiliki Accuracy tertinggi. Sehingga dapat dipastikan bahwa pemodelan Logistic Regression adalah pemodelan terbaik untuk dataset salju karena memiliki Accuracy tertinggi yaitu dengan nilai 84,60%.