LAPORAN TUGAS BESAR TAHAP II

MATA KULIAH PEMBELAJARAN MESIN SEMESTER GENAP 2021

Alif Ranadian Nadhifah
1301184255

Ibnu Muzakky M. Noor
1301184329

1. Formulasi Masalah

Diberikan sebuah dataset yaitu data salju_train.csv dan salju_test.csv. Dataset ini menyimpan informasi harian mengenai kondisi dan cuaca di suatu daerah untuk dibangun sebuah model clustering untuk memprediksi apakah pada hari besok akan turun salju atau tidak. Dataset tersebut masing-masingnya berupa himpunan data dengan 23 kolom atribut dan 1 kolom label diataranya: id, Tanggal, KodeLokasi, SuhuMin, SuhuMax, Hujan, Penguapan, SinarMatahari, ArahAnginTerkencang, KecepatanAnginTerkencang, ArahAngin9am, ArahAngin3pm, Kelembaban3pm, Tekanan9am, Tekanan3pm, Awan9am, Awan3pm, Suhu9am, Suhu3pm, BersaljuHariIni, dan BersaljuBesok.

2. Persiapan Data

Persiapan data bertujuan untuk membantu pengguna dalam melakukan preprocessing yang sesuai dan teknik analisis data yang digunakan agar data yang digunakan lebih berkualitas untuk dipakai dalam memprediksi nantinya. Pada persiapan data, hal yang dilakukan adalah mengetahui karakteristik data seperti mengetahui detail tipe data pada setiap data kolom, mengecek dan menghitung apakah data memiliki nilai kosong atau tidak, dan sebagainya.

Sebelum dataset diolah, langkah pertama yang harus dilakukan adalah kenali dahulu datasetnya. Pada gambar berikut menunjukkan code untuk mengetahui info terkait dataset.

```
datasalju.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 109095 entries, 0 to 109094
Data columns (total 23 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   Tanggal                  109095 non-null  object
 1   KodeLokasi               109095 non-null  object
 2   SuhuMin                  107973 non-null  float64
 3   SuhuMax                  108166 non-null  float64
 4   Hujan                    106664 non-null  float64
 5   Penguapan                62071 non-null   float64
 6   SinarMatahari            56716 non-null   float64
 7   ArahAnginTerkencang      101351 non-null  object
 8   KecepatanAnginTerkencang 101399 non-null  float64
 9   ArahAngin9am             101172 non-null  object
 10  ArahAngin3pm             105898 non-null  object
 11  KecepatanAngin9am        107742 non-null  float64
 12  KecepatanAngin3pm        106792 non-null  float64
 13  Kelembaban9am            107093 non-null  float64
 14  Kelembaban3pm            105721 non-null  float64
 15  Tekanan9am               97768 non-null   float64
 16  Tekanan3pm               97787 non-null   float64
 17  Awan9am                  67251 non-null   float64
 18  Awan3pm                  64624 non-null   float64
 19  Suhu9am                  107755 non-null  float64
 20  Suhu3pm                  106397 non-null  float64
 21  BersaljuHariIni          106664 non-null  object
 22  BersaljuBesok            106664 non-null  object
dtypes: float64(16), object(7)
memory usage: 20.0+ MB
```

Terlihat bahwa dataset salju memiliki atribut dan label kolom diantaranya KodeLokasi, SuhuMin, SuhuMax, Hujan, Penguapan, SinarMatahari, ArahAnginTerkencang, KecepatanAnginTerkencang, ArahAngin9am, ArahAngin3pm, Kelembaban3pm, Tekanan9am, Tekanan3pm, Awan9am, Awan3pm, Suhu9am, Suhu3pm, BersaljuHariIni, BersaljuBesok. Namun, dari 24 kolom tersebut, ada beberapa kolom yang memiliki nilai kosong atau null (missing value). Karena terdapat data missing value maka yang harus dilakukan dalam melakukan preprocessing data adalah drop data tersebut atau menggantinya dengan nilai modus dari kolom data itu. Hal ini dilakukan karena akan berakibat pada keberhasilan model yang dibangun nantinya. Namun dalam percobaan yang kami kerjakan, kami melakukan pengisian nilai missing value dengan nilai modus dari masing-masing kolom atribut terkait nilai missing value itu sendiri.

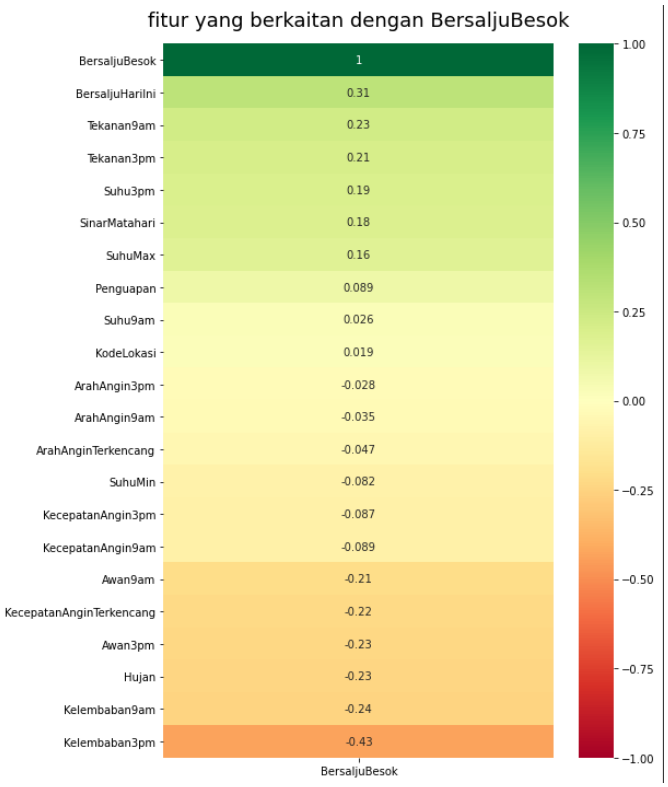
```
cols = ['Tanggal', 'KodeLokasi', 'SuhuMin', 'SuhuMax', 'Hujan', 'Penguapan',
        'SinarMatahari', 'ArahAnginTerkencang', 'KecepatanAnginTerkencang',
        'ArahAngin9am', 'ArahAngin3pm', 'KecepatanAngin9am',
        'KecepatanAngin3pm', 'Kelembaban9am', 'Kelembaban3pm', 'Tekanan9am',
        'Tekanan3pm', 'Awan9am', 'Awan3pm', 'Suhu9am', 'Suhu3pm',
        'BersaljuHariIni', 'BersaljuBesok']

for col in cols:
    datasalju[col] = datasalju[col].fillna(datasalju[col].mode()[0])
```

Untuk mengetahui kolom apa saja yang akan ataupun tidak digunakan dalam pembangunan model, maka dilakukan pemvisualisasian data menggunakan fungsi heatmap. Namun sebelum memvisualisasikan data ke dalam bentuk heatmap, perlu dilakukan encode data bertipe object menjadi data yang terdiri dari data bertipe integer. Hal ini dilakukan karena heatmap hanya menampilkan data yang bertipe integer ataupun float. Sedangkan dataset masih memiliki 7 kolom yang bertipe object, yaitu: ‘Tanggal’, ‘KodeLokasi’, ‘ArahAnginTerkencang’, ‘ArahAngin9am’, ‘ArahAngin3pm’, ‘BersaljuHariIni’, ‘BersaljuBesok’. Namun, untuk kolom atribut Tanggal tidak akan dipakai dalam membangun model nantinya sehingga tidak perlu dilakukan encode data value pada kolom atribut Tanggal. Berikut ini merupakan code dan hasil dari encode data yang bertipe object:

```
encode_dfs = {"KodeLokasi": {"C1":0, "C2":1, "C3":2, "C4":3, "C5":4, "C6":5, "C7":6, "C8":7, "C9":8, "C10":9,
                             "C11":10, "C12":11, "C13":12, "C14":13, "C15":14, "C16":15, "C17":16, "C18":17, "C19":18, "C20":19,
                             "C21":20, "C22":21, "C23":22, "C24":23, "C25":24, "C26":25, "C27":26, "C28":27, "C29":28, "C30":29,
                             "C31":30, "C32":31, "C33":32, "C34":33, "C35":34, "C36":35, "C37":36, "C38":37, "C39":38, "C40":39,
                             "C41":40, "C42":41, "C43":42, "C44":43, "C45":44, "C46":45, "C47":46, "C48":47, "C49":48},
              "ArahAnginTerkencang": {"E":0, "ENE":1, "ESE":2, "N":3, "NE":4, "NNE":5,
                                      "NNW":6, "NW":7, "S":8, "SE":9, "SSE":10,
                                      "SSW":11, "SW":12, "W":13, "WNW":14, "WSW":15},
              "ArahAngin9am": {"E":0, "ENE":1, "ESE":2, "N":3, "NE":4, "NNE":5,
                               "NNW":6, "NW":7, "S":8, "SE":9, "SSE":10,
                               "SSW":11, "SW":12, "W":13, "WNW":14, "WSW":15},
              "ArahAngin3pm": {"E":0, "ENE":1, "ESE":2, "N":3, "NE":4, "NNE":5,
                               "NNW":6, "NW":7, "S":8, "SE":9, "SSE":10,
                               "SSW":11, "SW":12, "W":13, "WNW":14, "WSW":15},
              "BersaljuHariIni": {"Ya":0 , "Tidak":1},
              "BersaljuBesok": {"Ya":0 , "Tidak":1}}
datasalju.replace(encode_dfs, inplace=True)
```

Fungsi dari memvisualisasikan heatmap adalah tak lain untuk memperlihatkan seberapa erat hubungan antar variabel kolom ataupun antara variabel kolom dengan label.



Pada gambar diatas terlihat bahwa, jika angka menunjukkan mendekati angka 0 (semakin menunjuk ke warna hijau tua), maka ini mengartikan hubungan/korelasi antar kolom terlalu jauh. Namun sebaliknya, bila angka menunjukkan mendekati angka 1 (semakin menunjuk ke warna orange gelap), maka ini mengartikan hubungan/korelasi hampir saling berkaitan. Dari gambar tersebut, terlihat bahwa 6 variabel kolom teratas yang memiliki keterkaitan erat dengan label data ‘BersaljuBesok’, karena memiliki nilai tertinggi adalah kolom 'SuhuMax', 'SinarMatahari', 'Tekanan9am', 'Tekanan3pm', 'Suhu3pm', 'BersaljuHariIni'. Sehingga nantinya pada saat pemodelan data yang dipakai hanya dataset salju yang memiliki 6 kolom atribut itu saja.

3. Pemodelan

Pada pemodelan pertama ini, model yang akan dipakai adalah model Supervised Learning yakni Classification: Decision Tree. Pertama yang dilakukan adalah men-split data fitur dan data label pada kedua dataset, salju_train dan salju_test.

```
[ ]  cols = ['SuhuMax', 'SinarMatahari', 'Tekanan9am', 'Tekanan3pm', 'Suhu3pm', 'BersaljuHariIni',]
     #Train
     X_train = datasalju[cols].to_numpy()
     y_train = datasalju['BersaljuBesok'].to_numpy()

     #Test
     X_test = datasalju_test[cols].to_numpy()
     y_test = datasalju_test['BersaljuBesok'].to_numpy()
```

Terdapat data kolom yang di ambil berdasarkan perhitungan fungsi korelasi di sebelumnya dan di inisialisasikan pada variable cols. Lalu memisahkan/men-split masing-masing dataset train dan dataset test menjadi dua bagian yaitu variabel untuk menempatkan data label dan variabel untuk menempatkan data fitur.

Setelah itu, inisialisasi variabel yang diperlukan dalam pembangunan model decision tree dan melakukan pelatihan model yang akan dibangun.

```
# membuat model Decision Tree
tree_model = DecisionTreeClassifier()

# melakukan pelatihan model terhadap data
tree_model.fit(X_train, y_train)
```

4. Eksperimen

Pada tahap eksperimen dilakukan eksekusi data yang diproses sebelumnya terhadap model yang telah dibangun dengan menggunakan library decision tree. Menguji data berdasarkan hasil akurasi model klasifikasi. Berikut hasil dari eksperimen yang dibangun:

```
[ ]  #0.0 "Ya", 1.0 "Tidak"
     y_pred = tree_model.predict(X_test)
     report = classification_report(y_test, y_pred)
     print(report)
```

```
                precision    recall  f1-score   support

           0.0       0.41      0.45      0.43      3939
           1.0       0.84      0.82      0.83     14243

      accuracy                           0.74     18182
     macro avg       0.63      0.63      0.63     18182
  weighted avg       0.75      0.74      0.74     18182
```

5. Kesimpulan

Berdasarkan data dan formula masalah yang ada. tahap pertama yang harus dilakukan dalam meng-clusterisasi suatu data adalah melakukan explorasi data dan processing data yang bertujuan untuk membantu pengguna memilih preprocessing data yang sesuai dan teknik analisis data yang digunakan. Seperti mengetahui karakteristik data seperti mengetahui detail tipe data pada setiap data kolom, mengecek dan menghitung apakah data memiliki nilai kosong atau tidak, dan semacam nya.

Pada tahap selanjutnya adalah melakukan pemodelan Supervised Learning yakni Classification: Decision Tree. dengan melakukan dilakukan adalah men-split data fitur dan data label pada kedua dataset, salju_train dan salju_test. Lalu melakukan inisialisasi variabel yang diperlukan dalam pembangunan model decision tree dan melakukan pelatihan model yang akan dibangun. Hal ini bertujuan untuk memudahkan dalam proses eksperimen/evaluasi model.

Lalu tahapan terakhir adalah melakukan pengujian atau eksperimen terhadap model yang dibangun dengan menggunakan library decision tree dan menguji data berdasarkan hasil akurasi model klasifikasi. Berdasarkan hasil eksperimen yang dilakukan bahwa hasil akurasi yang didapatkan berdasarkan model yang dibangun terhadap dataset salju_train sebesar 0.99. Namun hasil akurasi ini menjadi berbeda jika dilakukan prediksi terhadap dataset salju_test menggunakan model yang telah dibangun. Hasil akurasi yang didapat menjadi sebesar 0.74. Hasil akurasi ini menjadi berbeda dikarenakan, model yang telah dibangun sebelumnya menjadi berkurang keakuratannya ketika ia menerima dataset baru yaitu salju_test untuk diprediksi. Sedangkan sebelumnya, model belum pernah melakukan prediksi menggunakan dataset baru tersebut.

```
from sklearn.metrics import accuracy_score
y_pred_train = tree_model.predict(X_train)
print("Accuracy Score Data Train", accuracy_score(y_train, y_pred_train))
print("Accuracy Score Data Test", accuracy_score(y_test, y_pred))
```

```
Accuracy Score Data Train 0.9933360832302122
Accuracy Score Data Test 0.7391926080739193
```

6. Link Presentasi Youtube

https://youtu.be/B4QC5e3I6PI