Penerapan Machine Learning Automated Tools TPOT pada task dalam Dataset weatherAUS

Tugas Besar

Kelas MK Pembelajaran Mesin Lanjut (CII3L3)

Oleh :

I Nengah Dharma Pradnyandita 1301180296

Ryan Adeputra Sutopo 1301180297

Program Studi Sarjana Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2021

DAFTAR ISI

1. Formulasi masalah

Pada tugas besar ini akan dilakukan penyelesaian masalah dari sebuah dataset weatherAUS.csv. Masalah yang ingin diselesaikan adalah membuat prediksi potensi turunnya hujan pada besok hari. Model Automated Machine Learning yang digunakan adalah TPOT. Pada model ini akan dilakukan pelatihan dengan datasset, untuk menemukan solusi terbaik. Pada pemoresan data, akan dilakukan beberapa langkah untuk mengolah data agar dapat diterapkan dalam model Automated Machine Learning. Penggunaan model TPOT didasari dari beberapa penelitian sebelumnya yang memberikan performansi yang baik dalam melakukan pemodelan dari beberapa dataset. Pada penelitian ini diharapkan model memiliki akurasi lebih dari 80%.

2. Eksplorasi dan persipan data

Eksplorasi data dilakukan dalam melakukan pendekatan analisis untuk dataset dengan membuat gambaran dari keseluruhan data sehingga dapat untuk dipahami. Dalam proses ini karakteristik data akan dipelajari untuk melihat bagian yang akan di proses. Pada persiapan data, proses pre-processing dilakukan untuk mengolah data agar dapat menghasilkan output terbaik. Berikut langkah langkah yang dilakukan:

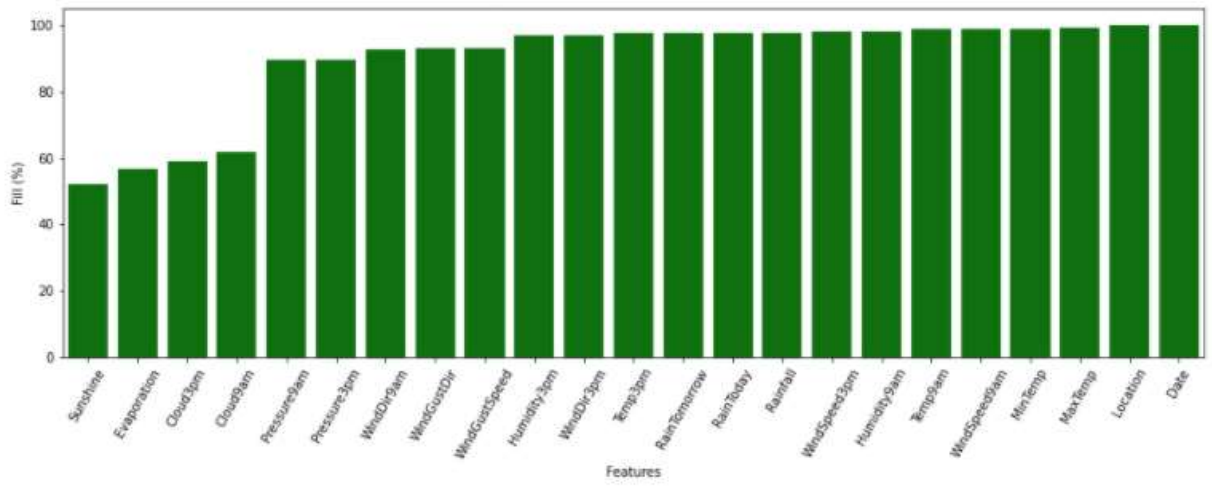
2.1. Eksplorasi Data

Pada tahap awal pemrosesan data, eksplorasi dilakukan untuk mengetahui karakteristik dari data. Dalam tahap ini dilakukan pencarian informasi dari data dan visualisasi data

```
: 1 data.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Date           145460 non-null  object
 1   Location       145460 non-null  object
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object
 10  WindDir3pm     141232 non-null  object
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object
 22  RainTomorrow   142193 non-null  object
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Gambar 2.1 Informasi kolom dan tipe data dari dataset weatherAUS.csv

Pada Gambar 1.1 diperlihatkan tipe data dari setiap atribut, dari informasi ini akan ditentukan tipe data yang dapat digunakan dalam proses pemodelan. Tidak semua tipe data dapat diproses oleh karena itu akan dilakukan perubahan tipe data pada langkah selanjutnya. Sebelum menentukan atribut yang akan digunakan dalam pemodelan, pada eksplorasi akan dilakukan visualisasi untuk mengetahui persentase data tidak *null* pada setiap atribut yang memiliki data tidak *null.*



Gambar 2.2 persentase data tidak null pada setiap atribut

Terlihat bahwa atribut yang memiliki data tidak null dengan persentase terendah adalah "*Suntime*" dan atribut yang memiliki persentase tertinggi data tidak *null* adalah Date. Pada proses pencarian atribut sesuai dilakukan dengan menggunakan korelasi antar atribut. Metode korelasi yang digunakan adalah Pearson Correlation dengan rumus sebagai berikut:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (1)$$

Keterangan :

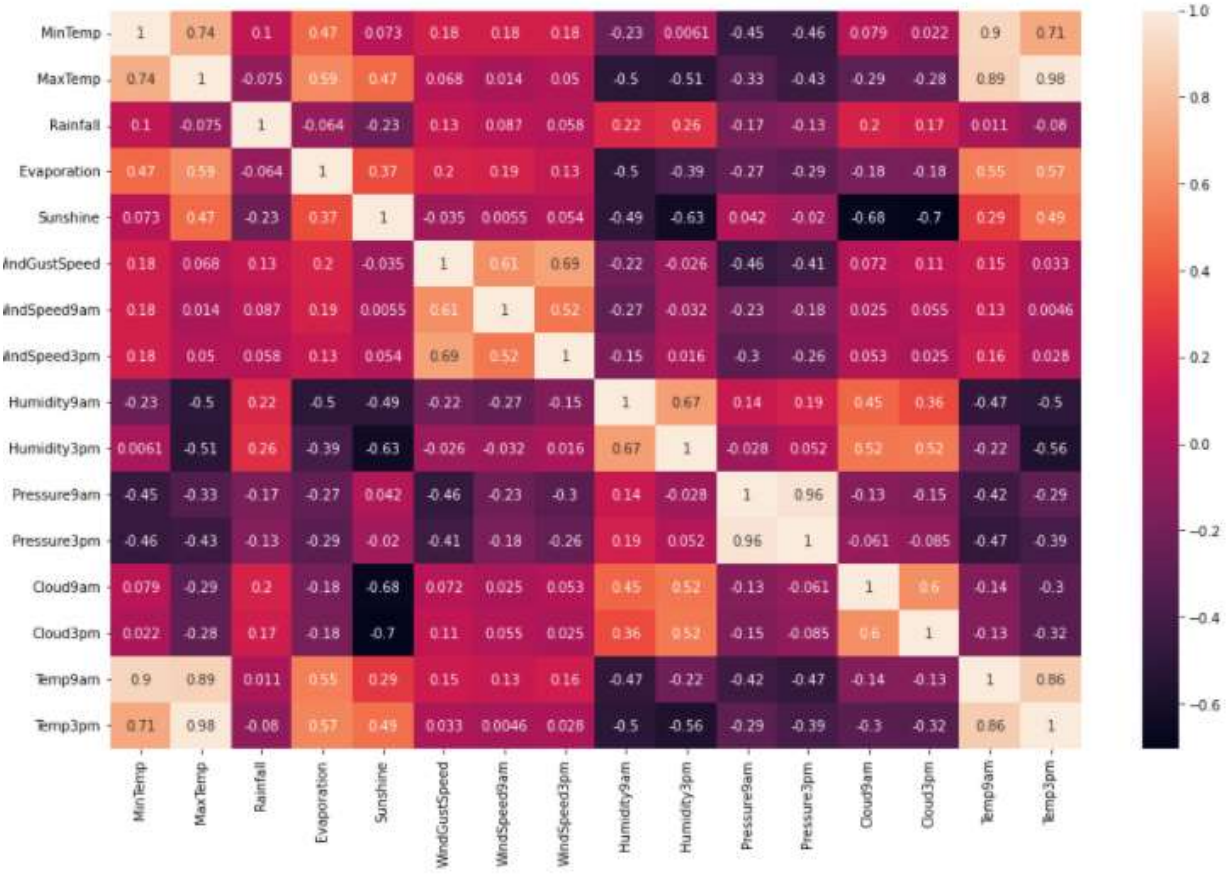
r = koefisien Pearson Correlation

x = nilai data pada set pertama

y = nilai data pada set kedua

n = banyaknya nilai

Metode korelasi tersebut kemudian diterapkan pada dataset dan divisualisaikan dengan menggunakan *library Seaborn*, berikut hasil dari visualisasi korelasi antara data:



Gambar 2.3 : korelasi antar atribut dalam dataset yang dihitung menggunakan Pearson Correlation

Pada korelasi tersebut hanya data yang bertipe numerical yang dimasukan kedalam perasamaan pearson correlation, sedangkan data katagorikal tidak. Terlihat pada visualisasi data, jika kotak semakin terang maka korealsi antar data adalah berkorelasi positif dan nilai semakin mendekati 1. Pada tahap praproses atribut yang tidak memiki korelasi yang signifikan akan di *drop.*

```
1 data.drop(['Date'], axis=1, inplace=True)
2 data.drop(['Location'], axis=1, inplace=True)
3 data.info()
```

Gambar 2.4 atribut yang tidak memiliki korelasi di drop.

2.2. Drop Duplicate Data

Pada dataset, terkadang ditemukan data yang memiliki nilai sama, sehingga akan dilakukan penghapusan terhadap data yang sama, agar terhindar dari redudansi data dan mempercepat performansi dalam pemodelan data.

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidi
0	13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW	20.0	24.0	
1	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW	4.0	22.0	
2	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW	19.0	26.0	
3	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E	11.0	9.0	
4	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW	7.0	20.0	
...	...	...	...	...	...	...	...	...	...	...	...	
145225	2.8	23.4	0.0	NaN	NaN	E	31.0	SE	ENE	13.0	11.0	
145226	3.6	25.3	0.0	NaN	NaN	NNW	22.0	SE	N	13.0	9.0	
145227	5.4	26.9	0.0	NaN	NaN	N	37.0	SE	WNW	9.0	9.0	
145228	7.8	27.0	0.0	NaN	NaN	SE	28.0	SSE	N	13.0	7.0	
145229	14.9	NaN	0.0	NaN	NaN	NaN	NaN	ESE	ESE	17.0	17.0	

145230 rows × 21 columns

Gambar 2.5 hasil dari drop data yang duplikat.

2.3. Drop Data yang kosong (Null data)

Pada hasil eksplorasi terlihat ada beberapa atribut yang memiliki nilai kosong (*null*) sehingga perlu dilakukan praproses berupa menghilangkan atribut. Pertimbangan menghilangkan atribut data dilakukan dengan menghitung persentase data yang bernilai kosong pada setiap atribut. Jika persentase data yang bernilai kosong pada atribut lebih dari 30% maka atribut akan didrop. Data yang tidak di drop akan digunakan dalam proses pemodelan.

	Total	Percent
Sunshine	69606	0.479281
Evaporation	62561	0.430772
Cloud3pm	59128	0.407134
Cloud9am	55658	0.383240
Pressure9am	14835	0.102148
Pressure3pm	14798	0.101894
WindGustSpeed	10033	0.069084
Humidity3pm	4277	0.029450
Temp3pm	3379	0.023267
Rainfall	3076	0.021180
WindSpeed3pm	2832	0.019500
Humidity9am	2424	0.016691
Temp9am	1537	0.010583
WindSpeed9am	1537	0.010583
MinTemp	1255	0.008641
MaxTemp	1037	0.007140
WindGustDir	0	0.000000
WindDir9am	0	0.000000
WindDir3pm	0	0.000000
RainToday	0	0.000000
RainTomorrow	0	0.000000

Gambar 2.6 persentase data yang kosong pada masing masing atribut

Pada gambar 2.6 terlihat beberapa atribut memiliki persentase data kosong yang lebih dari 30%, dan akan dilakukan drop pada atribut tersebut. Selain atribut yang dihilangkan, data yang memiliki nilai kosong pada setiap atribut akan diisi dengan nilai terdekat pada data tersebut, pengisian data

yang kosong dilakukan dengan model K-Nearest Neighbors. Hasil dari pemrosesan data menjadi seperti berikut:

```
RainTomorrow     0
WindSpeed9am     0
MaxTemp          0
Rainfall         0
WindGustDir      0
WindGustSpeed    0
WindDir9am       0
WindDir3pm       0
WindSpeed3pm     0
RainToday        0
Humidity9am      0
Humidity3pm      0
Pressure9am      0
Pressure3pm      0
Temp9am          0
Temp3pm          0
MinTemp          0
dtype: int64
```

Gambar 2.7 banyaknya nilai null pada atribut yang dipertahankan

Pada gambar 2.7 ,dilakukan pemeriksaan nilai null pada masing masing atribut dan hasilnya setiap atribut sudah tidak memiliki nilai null. Atribut yang terdapat pada gambar 2.6 adalah atribut yang dipertahankan untuk melakukan pemodelan.

2.4. Drop Outliers

Data outlier adalah data yang memiliki persebaran sangat jauh dari rata-rata data. Data ini dapat menyebabkan performansi dari pemodelan berkurang. Namun pada dataset ini data outliers memiliki jumlah yang tidak terlalu banyak, sehingga jika dilakukan penghapusan pada data tersebut tidak akan berdampak signifikan terhadap hasil pemodelan. Pada tugas kali ini data outliers akan dihilangkan.

```
jumlah data dengan outliers (145230, 17)
jumlah data tanpa outliers (134202, 17)
```

	MinTemp	MaxTemp	Rainfall	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	WindSpeed3pm	Humidity9am	Humidity3pm	Pr
0	13.4	22.9	0.6	13.0	44.0	13.0	14.0	20.0	24.0	71.0	22.0	
1	7.4	25.1	0.0	14.0	44.0	6.0	15.0	4.0	22.0	44.0	25.0	
2	12.9	25.7	0.0	15.0	46.0	13.0	15.0	19.0	26.0	38.0	30.0	
3	9.2	28.0	0.0	4.0	24.0	9.0	0.0	11.0	9.0	45.0	16.0	
4	17.5	32.3	1.0	13.0	41.0	1.0	7.0	7.0	20.0	82.0	33.0	
...	...	...	...	...	...	...	...	...	...	...	...	
134197	3.5	21.8	0.0	0.0	31.0	2.0	0.0	15.0	13.0	59.0	27.0	
134198	2.8	23.4	0.0	0.0	31.0	9.0	1.0	13.0	11.0	51.0	24.0	
134199	3.6	25.3	0.0	6.0	22.0	9.0	3.0	13.0	9.0	56.0	21.0	
134200	5.4	26.9	0.0	3.0	37.0	9.0	14.0	9.0	9.0	53.0	24.0	
134201	7.8	27.0	0.0	9.0	28.0	10.0	3.0	13.0	7.0	51.0	24.0	

134202 rows × 17 columns

Gambar 2.8 banyaknya nilai null pada atribut yang dipertahankan

Pada gambar 2.8 terlihat bahwa dataset memiliki 145230 data ketika masih terdapat outliers dan ketika data outliers dihilangkan menjadi 134202 data. Dari selisih tersebut diketahui bahwa terdapat 7,6% data outliers pada dataset. Persentase ini tidak terlalu signifikan terhadap pemodelan dari data.

2.5. Features Encoding

Pada tahap ini data yang bertipe categorical akan diencode menjadi bertipe numerical, sehingga pemodelan dari data dapat berjalan dengan baik.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145230 entries, 0 to 145229
Data columns (total 21 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   MinTemp        143975 non-null  float64
 1   MaxTemp        144193 non-null  float64
 2   Rainfall       142154 non-null  float64
 3   Evaporation    82669 non-null   float64
 4   Sunshine       75624 non-null   float64
 5   WindGustDir    135134 non-null  object
 6   WindGustSpeed  135197 non-null  float64
 7   WindDir9am     134894 non-null  object
 8   WindDir3pm     141232 non-null  object
 9   WindSpeed9am   143693 non-null  float64
 10  WindSpeed3pm   142398 non-null  float64
 11  Humidity9am    142806 non-null  float64
 12  Humidity3pm    140953 non-null  float64
 13  Pressure9am    130395 non-null  float64
 14  Pressure3pm    130432 non-null  float64
 15  Cloud9am       89572 non-null   float64
 16  Cloud3pm       86102 non-null   float64
 17  Temp9am        143693 non-null  float64
 18  Temp3pm        141851 non-null  float64
 19  RainToday      142154 non-null  object
 20  RainTomorrow   142147 non-null  object
dtypes: float64(16), object(5)
memory usage: 23.3+ MB
```

Sebelum Encode

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145230 entries, 0 to 145229
Data columns (total 21 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   MinTemp        143975 non-null  float64
 1   MaxTemp        144193 non-null  float64
 2   Rainfall       142154 non-null  float64
 3   Evaporation    82669 non-null   float64
 4   Sunshine       75624 non-null   float64
 5   WindGustDir    145230 non-null  int64
 6   WindGustSpeed  135197 non-null  float64
 7   WindDir9am     145230 non-null  int64
 8   WindDir3pm     145230 non-null  int64
 9   WindSpeed9am   143693 non-null  float64
 10  WindSpeed3pm   142398 non-null  float64
 11  Humidity9am    142806 non-null  float64
 12  Humidity3pm    140953 non-null  float64
 13  Pressure9am    130395 non-null  float64
 14  Pressure3pm    130432 non-null  float64
 15  Cloud9am       89572 non-null   float64
 16  Cloud3pm       86102 non-null   float64
 17  Temp9am        143693 non-null  float64
 18  Temp3pm        141851 non-null  float64
 19  RainToday      145230 non-null  int64
 20  RainTomorrow   145230 non-null  int64
dtypes: float64(16), int64(5)
memory usage: 23.3 MB
```

Sesudah Encode

Gambar 2.9 Perbandingan tipe data atribut yang sebelum dan sesudah di encode

Atribut yang diencode antara lain ’WindGustDir', ‘WindDir9am', 'WindDir3pm', 'RainToday', dan 'RainTomorrow'. Atribut ini diubah karena memiliki pengaruh terhadap pemodelan dataset.

2.6. Split Data

Pada tahap ini data yang telah melalui tahap pra proses akan dibagi menjadi dua, yaitu data train dan data test, data train digunakan untuk melatih model dan data test digunakan untuk menguji akurasi dari model yang dihasilkan. Pada data train dan data test, variabel x mewakili data non-label dan variabel y mewakili data label. Pembagian data train dan data test adalah 75% untuk data train dan 25% untuk data test.

```
#Data Split
X = data.drop('RainTomorrow',axis=1)
y = data['RainTomorrow']

X_train, X_test, y_train, y_test = train_test_split(X, y,train_size = 0.75, test_size = 0.25)
```

Gambar 2.10 Pembagian data train dan data test

2.7. Scaling Data

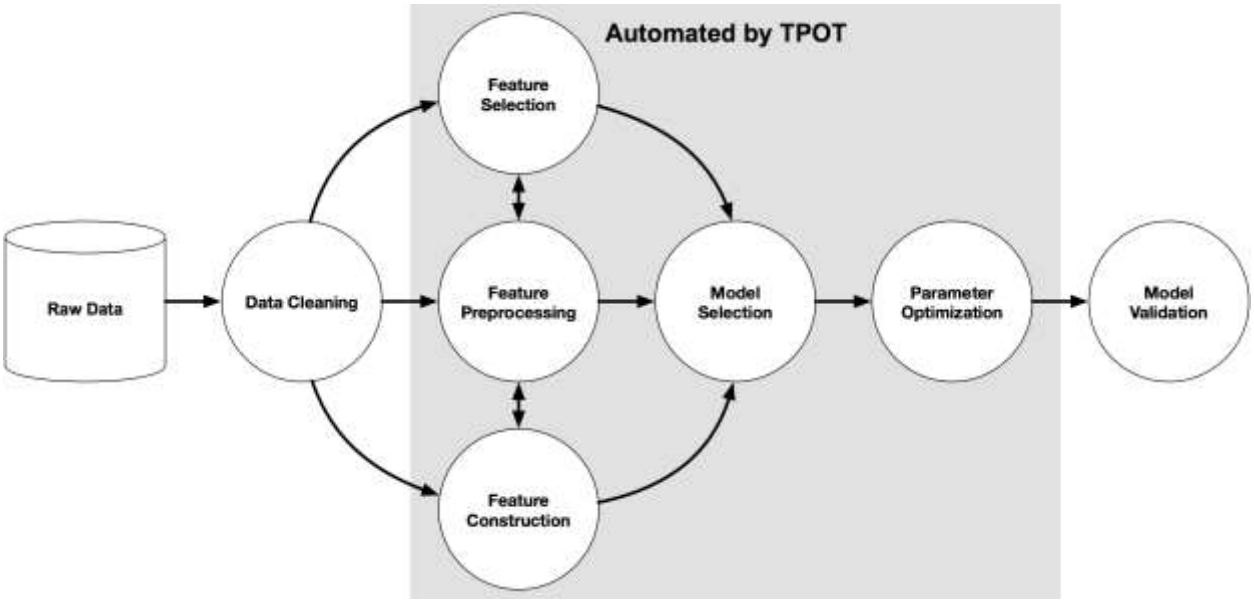
Scaling data dilakukan dengan menyamakan skala pada setiap atribut. hal ini agar model dapat bekerja dengan maksimal, metode scaling yang digunakan adalah *StandardScaler* yang bertujuan untuk membuat rata-rata 0 dan variansi 1. Berikut penerapan metode *StandardScaler*:

```
#Data Scalling
ss = StandardScaler()
X_train_scaled = ss.fit_transform(X_train)
X_test_scaled = ss.transform(X_test)
```

Gambar 2.11 Penerapan StandardScaler

3. Pemodelan

Automated Machine Learning (AutoML) adalah sebuah teknik untuk secara otomatis menemukan model berperformas baik untuk tugas pemodelan perdiktif dengan sedikit keterlibatan pengguna. TPOT merupakan salah satu *library open source* untuk AutoML pada bahasa pemrograman Python. Pada tugas ini digunakan library TPOT untuk melakukan pemodelan dari data weatherAUS.csv. Model TPOT dapat digambarkan sebagai berikut:



Gambar 3.1 Overview of the TPOT Pipeline Search sumber: Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science, 2016.

Pada tahap awal pemodelan dilakukan pengubahan parameter pada TPOT, parameter yang diubah adalah generations yang diset dengan 5, population_size diset dengan 50, verbosity diset dengan 2,dan n_jobs di set dengan 1. Dengan menggunakan konfigurasi TPOT secara default:

```
Optimization Progress:   0%|          | 0/300 [00:00<?, ?pipeline/s]

Generation 1 - Current best internal CV score: 0.85615642726431

Generation 2 - Current best internal CV score: 0.8569909936414251

Generation 3 - Current best internal CV score: 0.8569909936414251

Generation 4 - Current best internal CV score: 0.8569909936414251

Generation 5 - Current best internal CV score: 0.8569909936414251

Best pipeline: XGBClassifier(input_matrix, learning_rate=0.1, max_depth=8, min_child_weight=11, n_estimators=100, n_jobs=1, sub
sample=0.9500000000000001, verbosity=0)

TPOTClassifier(generations=5,
               periodic_checkpoint_folder='/content/drive/MyDrive/Tugas Besar '
                                          'AML/TPOT awal',
               population_size=50, verbosity=2)
```

Gambar 3.2. Hasil pemodelan dengan menggunakan 5 generasi dan populasi 50.

Pada tahap ini didapatkan hasil evaluasi sebagai berikut :

```
print('accuration: ', tpot.score(X_test_scaled, y_test))
```

```
accuration:  0.8580668236416202
```

```
y_prediction = tpot.predict(X_test_scaled)
print('classification report: \n', classification_report(y_test, y_prediction))
```

```
classification report:
               precision    recall  f1-score   support

         0.0       0.88      0.95      0.91     26574
         1.0       0.74      0.49      0.59      6977

    accuracy                           0.86     33551
   macro avg       0.81      0.72      0.75     33551
weighted avg       0.85      0.86      0.85     33551
```

Gambar 3.3. Hasil perhitungan akurasi dari pemodelan menggunakan TPOT

Hasil akurasi dari model pada tahap awal adalah 86% , kemudian dilakukan validasi silang untuk memperkirakan keterampilan model TPOT pada data yang tidak terlihat dengan menggunakan metode *k-fold cross-validation* dengan 5 lipatan, berikut hasil dari validasi model:

```
accuration:  0.848030997354793
classification report:
               precision    recall  f1-score   support

         0.0       0.87      0.95      0.91     21343
         1.0       0.69      0.47      0.56      5498

    accuracy                           0.85     26841
   macro avg       0.78      0.71      0.73     26841
weighted avg       0.84      0.85      0.84     26841
```

```
accuration:  0.8344323981967885
classification report:
               precision    recall  f1-score   support

         0.0       0.87      0.92      0.90     21211
         1.0       0.63      0.50      0.56      5630

    accuracy                           0.83     26841
   macro avg       0.75      0.71      0.73     26841
weighted avg       0.82      0.83      0.83     26841
```

```
accuration:  0.8459388971684053
classification report:
               precision    recall  f1-score   support

         0.0       0.87      0.94      0.90     20930
         1.0       0.70      0.52      0.60      5910

    accuracy                           0.85     26840
   macro avg       0.79      0.73      0.75     26840
weighted avg       0.84      0.85      0.84     26840
```

```
accuration:  0.8514157973174367
classification report:
               precision    recall  f1-score   support

         0.0       0.87      0.95      0.91     21104
         1.0       0.73      0.48      0.58      5736

    accuracy                           0.85     26840
   macro avg       0.80      0.72      0.75     26840
weighted avg       0.84      0.85      0.84     26840
```

```
accuration:  0.865946348733234
classification report:
               precision    recall  f1-score   support

         0.0       0.89      0.95      0.92     21747
         1.0       0.71      0.49      0.58      5093

    accuracy                           0.87     26840
   macro avg       0.80      0.72      0.75     26840
weighted avg       0.86      0.87      0.86     26840
```

Gambar 3.4. Hasil validasi model dengan 5 lipatan

Pada gambar 3.4 diperlihatkan hasil akurasi model dari setiap lipatan memiliki nilai diatas 80%. Hasil rata-rata dari akurasi model adalah sebagai berikut:

```
akurasi tiap lipatan — [0.848030997354793, 0.8344323981967885, 0.84593889716840
53, 0.8514157973174367, 0.865946348733234]
akurasi rata-rata : 0.8491528877541314
```

Gambar 3.5. Hasil rata rata validasi model dengan 5 lipatan

Pada hasil validasi model didapatkan nilai rata-rata 0.8491528877541314

4. Eksperimen

Pada eksperimen, dilakukan perubahan terhadap parameter model. Parameter yang diubah adalah parameter yang diubah adalah generations yang diset dengan 10, population_size diset dengan 5, verbosity diset dengan 2, n_jobs di set dengan 1, random_state diset dengan 42, cv diset dengan 5, early_stop diset dengan 4 dan Konfigurasi operator yang digunakan adalah TPOT Light. Konfigurasi ini digunakan karena TPOT akan mencari pada rentang terbatas dari praprosesor, konstruktor fitur, penyeleksi fitur, model, dan parameter untuk menemukan serangkaian operator yang meminimalkan kesalahan prediksi model. Hanya operator yang lebih sederhana dan berjalan cepat yang akan digunakan dalam jalur pipa ini, jadi TPOT Light ini berguna untuk menemukan jalur pipa yang cepat dan sederhana untuk masalah klasifikasi atau regresi.

```
Optimization Progress:   0%|          | 0/55 [00:00<?, ?pipeline/s]

Generation 1 - Current best internal CV score: 0.8463900211685985

Generation 2 - Current best internal CV score: 0.8463900211685985

Generation 3 - Current best internal CV score: 0.8490725697007502

Generation 4 - Current best internal CV score: 0.849102374972988

Generation 5 - Current best internal CV score: 0.849102374972988

Generation 6 - Current best internal CV score: 0.8493904991851313

Generation 7 - Current best internal CV score: 0.8493904991851313

Generation 8 - Current best internal CV score: 0.8493904991851313

Generation 9 - Current best internal CV score: 0.8516457911065037

Generation 10 - Current best internal CV score: 0.8516457911065037

Best pipeline: KNeighborsClassifier(DecisionTreeClassifier(LogisticRegression(input_matrix, C=5.0, dual=False, penalty=l2), cri
terion=gini, max_depth=8, min_samples_leaf=4, min_samples_split=10), n_neighbors=21, p=2, weights=distance)

TPOTClassifier(config_dict='TPOT light', early_stop=4, generations=10,
               periodic_checkpoint_folder='/content/drive/MyDrive/Tugas Besar '
                                          'AML/TPOT Light',
               population_size=5, random_state=42, verbosity=2)
```

Gambar 4.1. Hasil pemodelan pada eksperimen

Pada pemodelan tersebut terlihat terdapat 10 generasi.

5. Evaluasi Eksperimen

Pada tahap evaluasi eksperimen dilakukan dengan menggunakan library sklearn.metrics.Pada evaluasi ini didapatkan hasil sebagai berikut:

```
classification report:
              precision    recall  f1-score   support

         0.0       0.87      0.96      0.91     26705
         1.0       0.73      0.46      0.56      6846

    accuracy                           0.85     33551
   macro avg       0.80      0.71      0.74     33551
weighted avg       0.84      0.85      0.84     33551
```

Gambar 5.1. Hasil evaluasi pada eksperimen 1

Pada hasil evaluasi dapat terlihat akurasi yang dicapai adalah 85%. kemudian dilakukan validasi silang untuk memperkirakan keterampilan model TPOT pada data yang tidak terlihat dengan menggunakan metode *k-fold cross-validation* dengan 5 lipatan. Penggunaan metode ini dilakukan untuk mengetahui rata-rata keberhasilan dari model dengan cara melakukan perulangan dengan mengacak atribut masukan sehingga model teruji untuk beberapa atribut input yang acak. berikut hasil dari validasi model:

```
accuration:  0.8491859468723222
classification report:
              precision    recall  f1-score   support

         0.0       0.87      0.96      0.91     21343
         1.0       0.73      0.42      0.53      5498

    accuracy                           0.85     26841
   macro avg       0.80      0.69      0.72     26841
weighted avg       0.84      0.85      0.83     26841

accuration:  0.8316754219291382
classification report:
              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90     21211
         1.0       0.63      0.47      0.54      5630

    accuracy                           0.83     26841
   macro avg       0.75      0.70      0.72     26841
weighted avg       0.82      0.83      0.82     26841

accuration:  0.8411698956780924
classification report:
              precision    recall  f1-score   support

         0.0       0.86      0.94      0.90     20930
         1.0       0.71      0.48      0.57      5910

    accuracy                           0.84     26840
   macro avg       0.79      0.71      0.74     26840
weighted avg       0.83      0.84      0.83     26840
```

```
curation:  0.8435543964232489
assification report:
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91     21104
         1.0       0.75      0.40      0.52      5736

  accuracy                           0.84     26840
 macro avg       0.80      0.68      0.72     26840
ighted avg       0.83      0.84      0.82     26840

curation:  0.8603576751117735
assification report:
              precision    recall  f1-score   support

         0.0       0.88      0.96      0.92     21747
         1.0       0.73      0.42      0.53      5093

  accuracy                           0.86     26840
 macro avg       0.80      0.69      0.73     26840
ighted avg       0.85      0.86      0.84     26840
```

Gambar 5.2. Hasil validasi eksperimen dengan menggunakan metode k-fold cross-validation 5 lipatan.

Rata - rata dari hasil akurasi pada validasi model adalah 85%. Dari hasil validasi menunjukan bahwa model telah mencapai target akurasi yang telah ditentukan yaitu diatas 80%.

6. Kesimpulan

Automated Machine Learning (AutoML) adalah sebuah teknik untuk secara otomatis menemukan model berperformas baik untuk tugas pemodelan perdiktif dengan sedikit keterlibatan pengguna. TPOT merupakan salah satu *library open source* untuk AutoML pada bahasa pemrograman Python. Model TPOT cukup mudah digunakan, dalam penerapannya dapat dilakukan dengan melakukan pemanggilan Library. Pada model ini, diperlukan adanya tahap pra proses data yang baik agar hasil pemodelan maksimal. Dari hasil evaluasi ekperimen, terlihat bahwa model TPOT memberikan akurasi yang baik yaitu 85%. Hasil akurasi dipengaruhi oleh proses pengubahan parameter pada model TPOT dan hasil pra praproses data. Diharapkan untuk penelitan selanjutnya proses pengubahan parameter model yang digunakan dapat dikembangkan lagi agar hasil akurasi yang dicapai menjadi maksimal.