Laporan Tugas Besar Pembelajaran Mesin Lanjut

Laporan ini dibuat untuk memenuhi tugas besar

Mata Kuliah Pembelajaran Mesin Lanjut

Disusun oleh:

Mohammad Daffa Haris 1301180355

Muhammad Rizky Irfansyah 1301183471

S1 INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2020

Daftar Isi

A. Formulasi Masalah

Pada tugas besar ini, masalah yang akan diselesaikan adalah membuat program Machine Learning dengan menerapkan AutoML yang dapat memprediksi turun atau tidaknya hujan disuatu daerah berdasarkan parameter-parameter yang mempengaruhi cuaca. Data yang digunakan adalah dataset weatherAUS.csv. Dalam melakukan evaluasi digunakan TPOT Score dari setiap pipeline yang dihasilkan. Harapan dari output accuracy >75%.

B. Import, Ekspolorasi, dan Persiapan Data

1. Import Data

   Dilakukan untuk meng-import dataset yang dibutuhkan. Dataset "weatherAUS.csv" disimpan dalam bentuk dataframe dengan bantuan library pandas.

```
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/weatherAUS.csv')
```

2. Eksplorasi data

   Dalam tugas besar ini didapatkan dataset berupa hasil observasi beberapa elemen yang memiliki pengaruh terhadap cuaca di suatu daerah, selain itu dataset ini juga memprediksi kemungkinan hujan untuk esok hari. Dataset ini bernama "weatherAUS.csv", dataset "weatherAUS.csv" memiliki 145.460 data. Dataset ini memiliki 20 fitur dan sebuah label. Adapun fitur-fitur dari dataset ini diantaranya adalah:

   1. MinTemp
   2. MaxTemp
   3. Rainfall
   4. Evaporation
   5. Sunshine
   6. WindGustDir
   7. WindGustSpeed
   8. WindDir9am
   9. WindDir3pm
   10. WindSpeed9am
   11. WindSpeed3pm
   12. Humidity9am
   13. Humidity3pm
   14. Pressure9am
   15. Pressure3pm
   16. Cloud9am
   17. Cloud3pm
   18. Temp9am
   19. Temp3pm
   20. RainToday

   Dan label dari dataset ini adalah RainTomorrow.

3. Pre-Processing Data

   Sebelum melakukan pemodelan diperlukan persiapan terhadap dataset "weatherAUS.csv". Persiapan data dilakukan agar pemodelan yang dihasilkan lebih optimal. Adapun tahapan yang dilakukan dalam pre-processing adalah:

   1. Data Cleaning

      Dilakukan pembersihan terhadap baris data yang tidak memiliki label. . Data yang tidak memiliki label dihapus karena tidak menghasilkan akurasi yang lebih baik.

```
#menghapus data yang tidak memiliki label
data = data[~data['RainTomorrow'].isnull()]
```

   2. Missing Value Handling

      Pada tahap ini missing values diatasi dengan cara melakukan imputasi nilai menggunakan mean dan modus. Imputasi missing values dengan menggunakan nilai modus dilakukan untuk fitur bertipe kategorikal, sedangkan untuk fitur bertipe kontinu menggunakan nilai mean. Sehingga sebelum melakukan penaganan terhadap missing value perlu dilakukan pembagian fitur berdasar tipenya, yaitu fitur bertipe kontinu dan fitur bertipe kategorikal.

```
catFeature = ['Location',
              'WindGustDir',
              'WindDir9am',
              'WindDir3pm',
              'RainToday']
for i in catFeature:
  mode = data[i].mode()
  data[i].fillna(mode[0], inplace = True)

numFeature = ['MinTemp',
              'MaxTemp',
              'Rainfall',
              'Evaporation',
              'Sunshine',
              'WindGustSpeed',
              'WindSpeed9am',
              'WindSpeed3pm',
              'Humidity9am',
              'Humidity3pm',
              'Pressure9am',
              'Pressure3pm',
              'Cloud9am',
              'Cloud3pm',
              'Temp9am',
              'Temp3pm']
for i in numFeature:
  mean = data[i].mean()
  data[i].fillna(mean, inplace = True)
```

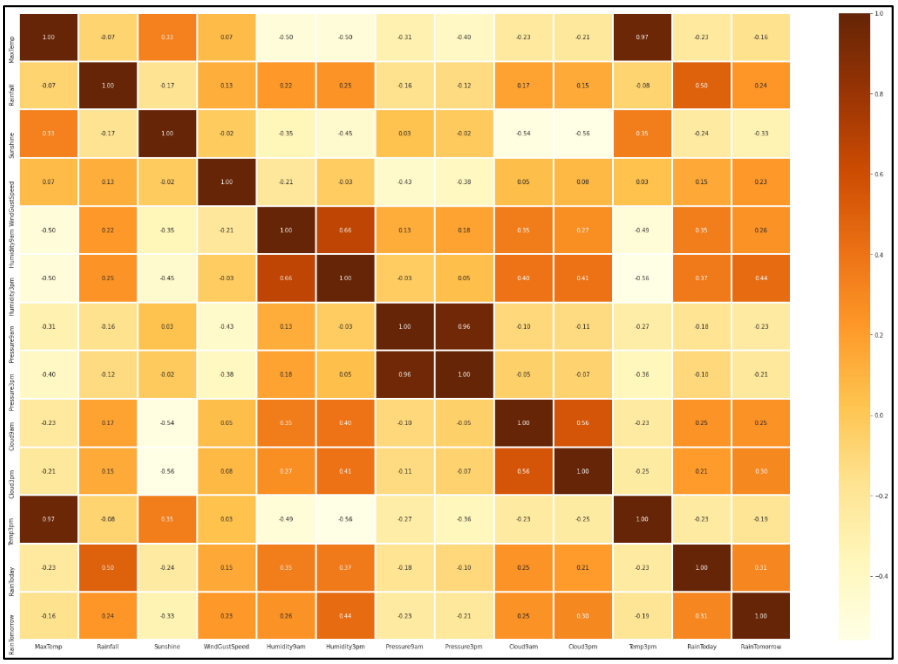
3. Transformasi Fitur Kategorikal

   Beberapa fitur pada dataset “weatherAUS.csv” bertipe kategorikal, maka dari itu untuk mempermudah proses komputasi dilakukan transformasi fitur menggunakan fungsi LabelEncoder yang disediakan oleh library Sklearn.

```
for i in catFeature:
  data[i] = LabelEncoder().fit_transform(data[i])
data['RainTomorrow'] = LabelEncoder().fit_transform(data['RainTomorrow'])
```

4. Feature Selection

   Feature Selection dilakukan untuk mengurangi kompleksitas fitur yang akan diolah pada tahapan modelling, sehingga diharapkan hasil modelling akan semakin optimal. Selain itu melakukan feature selection juga akan mempersingkat waktu pemrosesan TPOT. Untuk menentukan fitur yang akan di hapus digunakan correlation matrix atau matriks korelasi yang menggambarkan keterhubungan antara satu atribut dengan atribut lainnya. Semakin menjauhi nol (mendekati 1 atau -1) sebuah nilai korelasi antar atribut maka semakin berhubungan pula kedua atribut tersebut.



   Setelah dibuatnya correlation matrix menggunakan library Seaborn, didapatkan bahwa beberapa ftur memiliki nilai korelasi yang cukup rendah terhadap label.

Dalam eksperimen akan dilakukan penghapusan beberapa fitur mulai dari fitur yang memiliki korelasi dibawah 5% hingga dibawah 10%.

```
data.drop(['Location', 'MinTemp','Evaporation','WindGustDir','WindDir9am','WindDir3pm','WindSpeed9am',
           'WindSpeed3pm','Temp9am'], axis='columns', inplace=True)
#data.drop(['Location','WindGustDir','WindDir9am','WindDir3pm','Temp9am'], axis='columns', inplace=True)
```

5. Data Split

   Data Split merupakan tahapan untuk memisahakan kolom fitur dan dan kolom label dari sebuah dataset. Tahapan ini berguna untuk mempermudah proses modelling yang akan dilakukan. Selain itu dataset juga akan dipecah menjadi data test dan data train. Data train digunakan untuk melatih model yang telah dihasilkan, sedangkan data test digunakan untuk pengujian terhadap model yang telah dibuat. Dalam eksperimen ini pembagian data train dan data test juga berubah-ubah, perbandingan antara data train dan data test adalah 85%:15% dan 80%:20%. Untuk mempermudah proses data split digunakan function train_test_plit yang disediakan oleh library Sklearn.

```
label = data['RainTomorrow']
feature = data.drop('RainTomorrow', axis=1)
featureTrain, featureTest,labelTrain,labelTest = train_test_split(feature,label,
                                                    test_size = 0.15,random_state = 12)
```

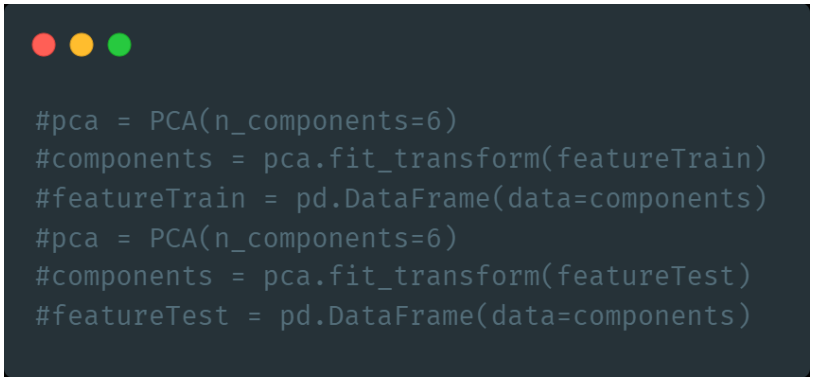
6. Data Scaling

   Data Scaling digunakan untuk membuat fitur pada dataset “weatherAUS.csv” berada pada rentang nilai yang sama. Data sacling diperlukan agar jarak perhitungan antar fitur tidak terlalu jauh. Dalam eksperimen ini dilakukan data scaling menggunakan dua metode yaitu MinMaxScaler dan StandardScaller, dimana kedua fungsi ini disediakan oleh library Sklearn.

```
featureTrain = featureTrain.drop('Date',axis=1)
featureTest = featureTest.drop('Date',axis=1)
#featureTrain = MinMaxScaler().fit_transform(featureTrain)
#featureTest = MinMaxScaler().fit_transform(featureTest)
featureTrain = StandardScaler().fit_transform(featureTrain)
featureTest = StandardScaler().fit_transform(featureTest)
```

7. Dimensionality Reduction

   Dalam eksperimen juga dilakukan Dimensionality Reduction. Dimensionality Reduction digunakan untuk meringkas fitur-fitur yang sudah dipilih menjadi fitur yang lebih sederhana, pada kasus kali ini fitur yang awalnya berjumlah 20 akan dirubah menjadi 6 fitur. Metode yang digunakan adalah Principal Component analysis (PCA). Metode ini dipilih karena mudah diimplementasikan. Dengan menggunakan metode ini semua fitur yang berguna akan tetap berpengaruh, walaupun dimensi dari dataset sudah dikecilkan. Metode PCA ini juga disediakan oleh library Sklearn. Diharapkan dengan dilakukan Dimensionality Reduction diharapkan proses komputasi dapat berjalan lebih cepat.



```
#pca = PCA(n_components=6)
#components = pca.fit_transform(featureTrain)
#featureTrain = pd.DataFrame(data=components)
#pca = PCA(n_components=6)
#components = pca.fit_transform(featureTest)
#featureTest = pd.DataFrame(data=components)
```

C. Pemodelan

Tool AutoML yang digunakan pada pemodelan ini adalah TPOT. Tool ini dipilih karena cenderung mudah diimplementasikan. TPOT akan memilih algoritma pemodelan yang paling optimal beserta parameternya. TPOT akan mencoba sebuah pipeline dan mengevaluasi nilai keakurasiannya, lalu TPOT akan merubah step dalam pipeline secara random dan mencari nilai keakurasian yang paling optimum dalam sebuah dataset. Pipeline sendiri merupakan tahapan-tahapan yang dilalui untuk mendapatkan nilai akurasi dari sebuah model.

D. Eksperimen

Hyperparameter yang digunakan dalam eksperimen TPOT ini adalah adalah:

1. Verbosity = 2

   Verbosity merupakan intensitas komunikasi dari TPOT ketika sedang berjalan. Dipilihnya value 2 adalah agar muncul progess bar ketika TPOT sedang berjalan.

2. Generation = 3

   Generation merupakan jumlah generasi yang dibuat dalam mengoptimalkan pipeline yang dihasilkan. Dipilihnya value 3 karena besarnya dataset yang ada sehingga pembangkitan generasi tidak perlu terlalu banyak. Sehingga diharapkan proses sunning program ini tidak terlalu lama.

Selain menentukan hyperparameter juga dilakukan beberapa eksperimen dalam melakukan pre-processing dataset. Beberapa eksperimen yang dilakukan diantaranya adalah:

1. Running program #1

Metode Scaling	:	MinMaxScaler
Reduksi Jumlah Fitur	:	Dimensionality Reduction dengan metode PCA menjadi 6 fitur
Ukuran Data Test	:	15%
Ukuran Data Train	:	85%

2. Running program #2

Metode Scaling	:	StandardScaler
Reduksi Jumlah Fitur	:	Feature Selection, menghapus fitur yang memiliki nilai korelasi dibawah 10% terhadap label
Ukuran Data Test	:	15%
Ukuran Data Train	:	85%

3. Running program #3

Metode Scaling	:	MinMaxScaler
Reduksi Jumlah Fitur	:	Feature Selection, menghapus fitur yang memiliki nilai korelasi dibawah 10% terhadap label
Ukuran Data Test	:	15%
Ukuran Data Train	:	85%

4. Running program #4

Metode Scaling	:	StandardScaler
Reduksi Jumlah Fitur	:	Feature Selection, menghapus fitur yang memiliki nilai korelasi dibawah 5% terhadap label
Ukuran Data Test	:	20%
Ukuran Data Train	:	80%

E. Evaluasi

Dalam melakukan evaluasi digunakan TPOT Score dari setiap pipeline yang dihasilkan. Berikut ini merupakan hasil output yang didapatkan:

1. Running Program #1
   - Pipeline

```
Best pipeline: MLPClassifier(input_matrix, alpha=0.0001, learning_rate_init=0.001)
```

   - TPOT Score

```
tpot.score(featureTest, labelTest)
0.7836279244221482
```

2. Running Program #2
   - Pipeline

```
Best pipeline: RandomForestClassifier(input_matrix, bootstrap=True, criterion=gini, max_features=0.25, min_samples_leaf=4, min_samples_split=2, n_estimators=100)
```

   - TPOT Score

```
tpot.score(featureTest, labelTest)
0.8511416381452482
```

3. Running Program #3
   - Pipeline

```
Best pipeline: XGBClassifier(input_matrix, learning_rate=0.1, max_depth=8, min_child_weight=7, n_estimators=100, n_jobs=1, subsample=0.8, verbosity=0)
```

   - TPOT Score

```
tpot.score(featureTest, labelTest)
0.84392142153875
```

4. Running Program #4
   - Pipeline

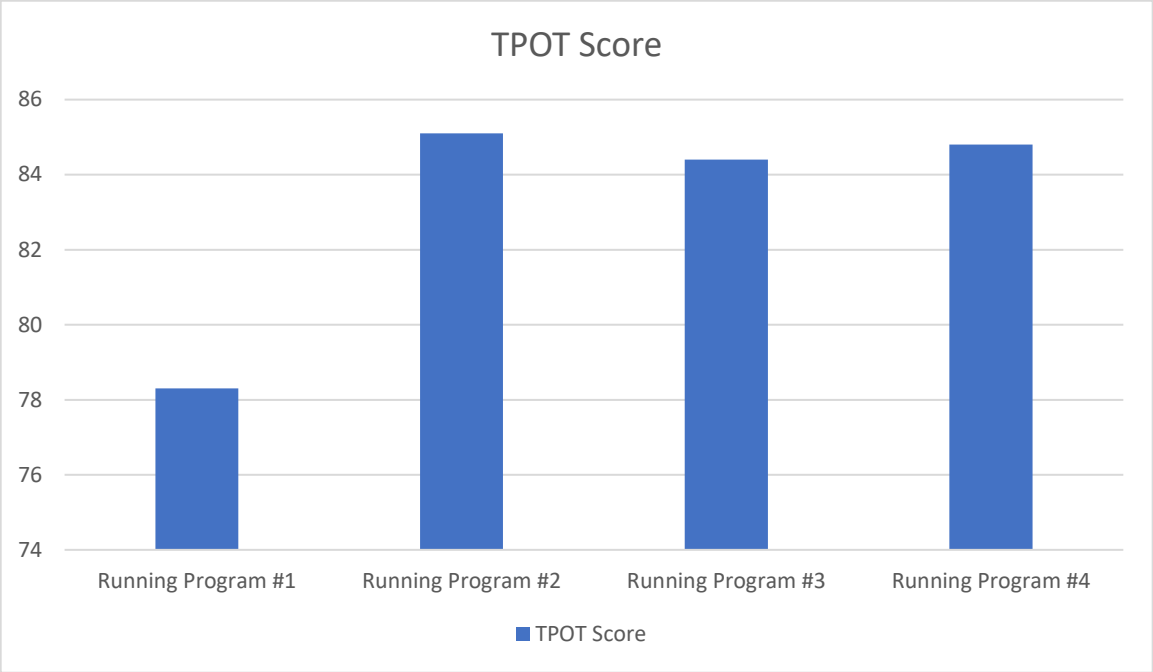
```
Best pipeline: GradientBoostingClassifier(input_matrix, learning_rate=0.1, max_depth=7, max_features=0.5, min_samples_leaf=16, min_samples_split=13, n_estimators=100, subsample=0.7000000000000001)
```

   - TPOT Score

```
tpot.score(featureTest, labelTest)
0.8484123914342979
```

F. Kesimpulan

Setelah dilakukan running program sebanyak 4 kali didapatkan hasil TPOT Score sebagai berikut:



Didapatkan TPOT Score tertinggi sebesar 85,1% yang didapatkan dari running program #2 dengan pipeline yang dihasilkan adalah:

```
Best pipeline: RandomForestClassifier(input_matrix, bootstrap=True, criterion=gini, max_features=0.25, min_samples_leaf=4, min_samples_split=2, n_estimators=100)
```

Sedangkan parameter yang digunakan adalah sebagai berikut:

- Hyperparameter TPOT: Verbosity = 2, Generation = 3
- PreProcessing:

Metode Scaling	:	StandardScaler
Reduksi Jumlah Fitur	:	Feature Selection, menghapus fitur yang memiliki nilai korelasi dibawah 10% terhadap label
Ukuran Data Test	:	15%
Ukuran Data Train	:	85%

Setelah dilakukan running sebanyak 4 kali running program didapatkan beberapa poin kesimpulan diantaranya adalah:

1. Penerapan Dimensionality Reduction dengan metode PCA justru menurunkan akurasi sebesar $\pm$ 5% dibandingkan running program yang lain. Kemungkinan besar hal ini terjadi karena dalam TPOT sendiri sebenarnya sudah dilakukan Dimensinality Reduction menggunakan

metode PCA sehingga apabila dilakukan Dimensinality Reduction ketika pre-processing data akan menyebabkan penurunan akurasi.

2. Penggunaan StandardScaler dan MinMaxScaler tidak memiliki perbedaan yang signifikan, padahal dalam metode scaling MinMaxScaler dipastikan setiap fitur tidak memiliki outlier/noise.