**LAPORAN TUGAS BESAR ADVANCE ML**

1. **Formulasi Masalah**

Masalah yang akan kita selesaikan menggunakan Machine Learning adalah memprediksi apakah hari esok akan hujan atau tidak berdasarkan data-data cuaca yang diberikan. Dataset yang kami gunakan adalah data weatherAUS.csv yang nantinya akan dibagi menjadi dua dataset yaitu data training dan data testing pada bagian *Data Eksploration and Cleansing*. Dataset yang akan dilakukan pemodelan menggunakan tools Machine Learning adalah Data Training, sedangkan Data Testing digunakan untuk mengecek keakuratan hasil pemodelan terhadap dataset awal.

Masalah lain yang dapat diselesaikan menggunakan Machine Learning terhadap Dataset weatherAUS.csv adalah memprediksi apakah hari itu akan turun hujan atau tidak. Hal ini didasari oleh variable-variabel yang terdapat dalam Dataset tersebut, seperti kecepatan angin dan kelembapan pada hari itu.

Menggunakan Machine Learning untuk dapat menyelesaikan masalah ini tentu saja memiliki dampak yang baik terhadap seseorang atau suatu kelompok yang memegang dataset tersebut. Hasil prediksi dapat digunakan untuk mengantisipasi keadaan, seperti contohnya kita dapat mengetahui perlukah kita untuk membawa payung atau tidak.

1. **Data Eksploration and Cleansing**

Pada bagian ini, mahasiswa diwajibkan untuk mengetahui informasi-informasi dari dataset. Memahami data merupakan hal yang penting untuk dilakukan sebelum melakukan pemodelan. Terdapat beberapa tahapan yang kami lakukan pada bagian ini.

* 1. Data Exploration/Understanding

Berikut merupakan gambaran dari dataset yang akan digunakan pada bagian eksplorasi data.

A screenshot of a computer

Description automatically generated

Perhatikan pada bagian yang digaris bawahi, dapat diketahui bahwa terdapat 145460 baris dan 23 kolom pada dataset ini. Dataset tersebut memiliki keterangan kolom sebagai berikut.

|  |  |
| --- | --- |
| Nama Kolom | Keterangan |
| Date | Tanggal observasi (objek Tanggal). |
| Location | Nama Lokasi pengujian |
| MinTemp | Suhu minimum dalam derajat celsius. |
| MaxTemp | Suhu maximum dalam derajat celsius. |
| Rainfall | Jumlah curah hujan yang tercatat untuk hari itu dalam mm. |
| Evaporation | Yang disebut penguapan panci Kelas A (mm) dalam 24 jam hingga 9 pagi. |
| Sunshine | Jumlah jam sinar matahari cerah dalam sehari. |
| WindGustDir | Arah hembusan angin terkuat dalam 24 jam hingga tengah malam. |
| WindGustSpeed | Kecepatan (km / jam) hembusan angin terkuat dalam 24 jam hingga tengah malam. |
| WindDir9am | Arah hembusan angin terkuat pada jam 9 pagi. |
| WindDir3pm | Arah hembusan angin terkuat pada jam 3 sore. |
| WindSpeed9am | Kecepatan angin (km / jam) rata-rata lebih dari 10 menit sebelum jam 9 pagi |
| WindSpeed3pm | Kecepatan angin (km / jam) rata-rata lebih dari 10 menit sebelum jam 3 sore. |
| Humidity9am | kelembaban (persen) pada jam 9 pagi. |
| Humidity3pm | kelembaban (persen) pada jam 3 sore. |
| Pressure9am | Tekanan atmosfer (hpa) berkurang hingga rata-rata permukaan laut pada pukul 9 pagi. |
| Pressure3pm | Tekanan atmosfer (hpa) berkurang hingga rata-rata permukaan laut pada pukul 3 sore. |
| Cloud9am | Bagian langit tertutup awan pada jam 9 pagi. Ini diukur dalam "oktas", yang merupakan satuan delapan. Ini mencatat berapa delapan ratus delapan puluh langit yang tertutup awan. Ukuran 0 menunjukkan langit cerah sepenuhnya sementara angka 8 menunjukkan bahwa langit benar-benar mendung. |
| Cloud3pm | Bagian langit tertutup awan pada jam 3 sore. Ini diukur dalam "oktas", yang merupakan satuan delapan. Ini mencatat berapa delapan ratus delapan puluh langit yang tertutup awan. Ukuran 0 menunjukkan langit cerah sepenuhnya sementara angka 8 menunjukkan bahwa langit benar-benar mendung. |
| Temp9am | Suhu (derajat C) pada jam 9 pagi. |
| Temp3pm | Suhu (derajat C) pada jam 3 sore. |
| RainToday | Yes jika curah hujan (mm) dalam 24 jam hingga 9 pagi melebihi 1 mm, jika tidak No. |
| RainTomorrow | Variabel target. Apakah besok hujan atau tidak. |

*Table Keterangan kolom*

Setelah itu kita identifikasi mana kolom dengan variable categorical, karena umumnya variable categorical memiliki value bertipe data string, sehingga akan terjadi eror ketika akan dilakukan operasi matematika atau fungsi aggregasi table. Kita tahu bahwa kolom yang merupakan varibel categorical dengan tipe data kolom tersebut string adalah kolom Date,Location, RainToday, RainTomorrow, WindGustDir, WinDir9am,dan WindDir3pm. Namun kolom Date dan Location tidak akan kami gunakan karena kedua kolom tersebut sifatnya index. Disini kami ubah(mapping) isi kolom tersebut berdasarkan kategori sebagai berikut.

|  |  |  |
| --- | --- | --- |
| Nama Kolom | Sebelum mapping | Setelah Mapping |
| RainToday, RainTomorrow | Yes | 1 |
| No | 0 |
| WindGustDir, WindDir9am,WindDir3pm | W | 0 |
| WNW | 1 |
| NW | 2 |
| NNW | 3 |
| N | 4 |
| NNE | 5 |
| NE | 6 |
| ENE | 7 |
| E | 8 |
| ESE | 9 |
| SE | 10 |
| SSE | 11 |
| S | 12 |
| SSW | 13 |
| SW | 14 |
| WSW | 15 |

*Table Keterangan mapping*

Maka hasil setelah mapping adalah sebagai berikut.

Table

Description automatically generated

Selanjutnya akan dilakukan pengecekan tipedata seluruh kolom, hasilnya sebagai berikut.

Table

Description automatically generated

Dapat kita lihat dari gambar diatas bahwa seluruh kolom kecuali kolom Date dan Location sudah bertipe float, sehingga data tersebut sudah dapat kita lihat deskripsi statistika singkatnya. Berikut merupakan Deskripsi Statistik singkat dari data set ini.’

A picture containing table

Description automatically generated

* 1. Data Cleansing

Data yang baik merupakan data yang bersih dari missing value dan outliers, Missing Value merupakan data yang hilang atau tidak diisi, umumnya missing value pada dataset berbentuk nan, namun ada juga missing value yang berbentuk karakter “-“. Sedangkan Outliers atau data pencilan merupakan data yang ganjil atau data yang aneh. Data yang memiliki Missing value dan outliers dapat berpengaruh terhadap pemodelan, sehingga perlu dilakukan handling terhadap dua kasus tersebut.

* Cek Missing Value Semua Kolom

Missing value dapat dilihat dengan menggunakan sintaks berikut.

Table

Description automatically generated

Dapat kita ketahui bahwa seluruh kolom kecuali kolom Date dan Location memiliki missing value, hal ini perlu kita atasi. Kami memilih untuk me*replace* missing value tersebut dengan nilai rata-rata dan modus variable tersebut. Untuk varibel kategorical, missing value di*replace* dengan modus, sedangkan variable numerik, missing value di*replace* dengan rata-rata(mean). Hal ini dilakukan untuk menjaga kestabilan distribusi data dari setiap variable.

Graphical user interface, text, application

Description automatically generated

Sebelum kita beralih ke pengecekan outliers, kita tahu bahwa terdapat kolom yang cara mengisinya berhubungan dengan kolom lain, yaitu kolom RainToday. Karena sebelumnya kita sudah sepakat mengatasi missing value dengan mean dan modus, maka ada kemungkinan bahwa nilai dari suatu baris pada Kolom Rainfall dan RainToday tidak match(lihat Kembali table keterangan kolom untuk RainToday). Maka dari itu kita perlu cek apakah ada data yang memiliki nilai Rainfall diatas 1 dengan RainToday 0(No), dan Rainfall dibawah 1 dengan RainToday 1(yes).

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Dari kedua gambar diatas, dapat kita ketahui bahwa terdapat data yang tidak match antara kolom Rainfall dan RainToday, yaitu data dengan Rainfall diatas 1.0 namun RainToday 0(No). maka dari itu hal ini perlu kita atasi dengan mengganti nilai RainToday yang 0(No) menjadi 1(yes) apabila rainfallnya diatas 1.0.

Text

Description automatically generated with medium confidence

* Cek Outliers semua kolom kecuali Date dan Location

Data Pencilan dapat kita lihat menggunakan boxplot.

Diagram

Description automatically generated with medium confidence

Chart, box and whisker chart

Description automatically generated with medium confidence

Bulatan-bulatan disekitar box merupakan data-data yang kurang dari batas bawah dan lebih dari batas atas pada variabel tersebut, hal ini dapat menjadi acuan dalam mendeteksi outliers.

Graphical user interface, text, table

Description automatically generated with medium confidence

Namun data-data tersebut belum dapat kita identifikasi sebagai outliers, karena bisa saja data-data tersebut memang kredibel. Mengingat bahwa kami belum memiliki kemampuan dalam mengidentifikasi bulatan-bulatan hitam disekitaran box itu merupakan data pencilan atau bukan, maka kami sepakat untuk membuang bulatan-bulatan hitam tersebut. Kami membuangnya berdasarkan interquartile range variable tersebut.

Text

Description automatically generated

Text, letter

Description automatically generated

Kedua gambar diatas merupakan fungsi untuk mengambil data-data yang dalam range >= batas bawah dan <= batas atas saja.

Perhatikan kedua gambar berikut.

Chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Melihat bentuk dari kedua boxplot diatas, jarak antara batas atas dan batas bawah variable tersebut sangatlah kecil, bahkan pada variable Rainfall batas bawah dan batas atasnya sama, sehingga kedua variable tersebut tidak memungkinkan untuk dibuang bulatan-bulatan hitamnya.

Variable lain yang tidak perlu dibuang bulatan-bulatan hitamnya adalah RainToday dan RainTomorrow, karena kedua variable tersebut hanya memiliki nilai 0 dan 1. Jadi tidak mungkin ada outliers.

Hasil akhir pada bagian ini adalah seluruh kolom kecuali kolom Date, Location, RainToday, RainTomorrow, Rainfall, dan Evaporation hanya memiliki data-data yang terdapat dalam range **batas bawah < data < batas atas** saja.

* 1. Feature Engineering dan drop unused column

Pada bagian ini, seluruh data dari kedua atribut akan dilakukan scalling. Scalling merupakan sebuah metode untuk mentranformasikan nilai data menjadi nilai dalam range tertentu. Disini saya menggunakan scalling menggunakan cara MinMax Normalization. MinMax Normalization akan mentransformasikan data kedalam range 0 – 1. Scalling dilakukan agar semua variable memiliki range nilai yang sama. Berikut merupakan cara perhitungan MinMax Normalization.

A picture containing text

Description automatically generated

Kolom yang akan di drop adalah Date dan Location, karena kami tidak membutuhkan variable tersebut untuk pemodelan karena kedua variable tersebut sifatnya hanyalah index. Berikut hasil scalling dan drop unused column.

Table

Description automatically generated

* 1. Data Splitting

Kami menggunakan rasio 66% untuk data train dan 33% untuk datatest.

* 1. Pemodelan

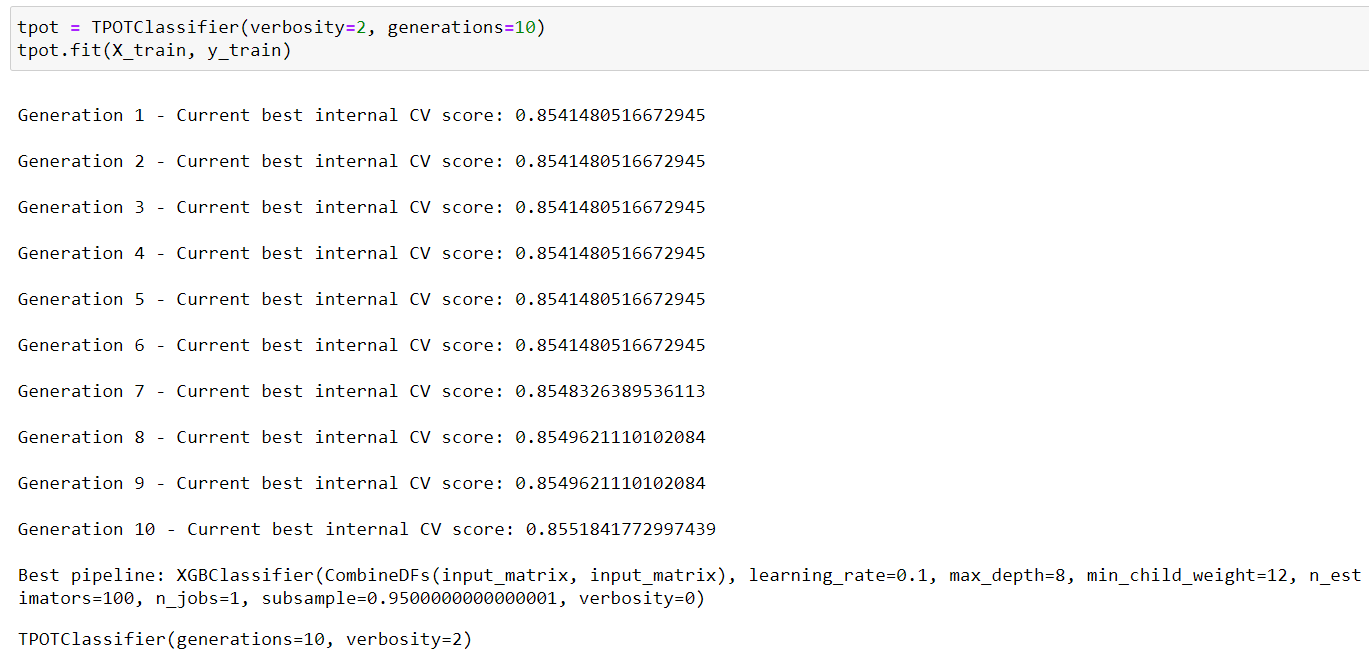
Kami menggunakan library TPOT pada bahasa pemrograman python karena selain gratis, instalasi TPOT juga sangat mudah. Selain itu TPOT juga dapat melakukan proses auto-ml secara mendalam, melakukan proses modelling dan klasifikasi secara otomatis, sehingga pada saat akhir dari proses TPOT kita dapat melakukan export kode dari proses klasifikasi yang paling optimal, dan dapat melihat metode klasfikasi apa yang menjadi metode paling optimal. Kita juga dapat memilih hyperparameter seperti generations dan max\_time agar dapat disesuaikan dengan kapasitas hardware yang digunakan.

Selain TPOT, kami juga menggunakan autokeras sebagai tools pembanding untuk bahan eksperimen kami.

Untuk TPOT, kami membatasi generasi maksimumnya sebanyak 10 generasi agar proses TPOT tidak terlalu lama. Sedangkan untuk autokeras, kami menggunakan max\_trials=20, yaitu jumlah model maksimum untuk dicoba. Namun proses autokeras dapat selesai sebelum menyentuh max\_trials. Selain max\_trials, kami juga menggunakan epochs=500 untuk melakukan train sebanyak jumlah epochs terhadap masing-masing model.

* 1. Eksperimen

Setelah TPOT dijalankan, akurasi yang didapatkan adalah sebesar 84.93726992712793%.





Metode klasifikasi yang optimal yang didapatkan dari hasil TPOT adalah metode XGBClassifier dengan hyperparameter sebagai berikut:

1. learning\_rate = 0.1
2. max\_depth = 8
3. min\_child\_weight = 12
4. n\_estimators = 100
5. n\_jobs = 1
6. subsample = 0.9500000000000001

Selanjutnya hasil TPOT kami export kedalam bahasa python, agar kedepannya jika jupyter notebook kami tutup, kami tidak perlu melakukan proses TPOT dari awal lagi.



Selanjutnya, kami membandingkan dengan hasil auto-ml dengan tools dari autokeras.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application

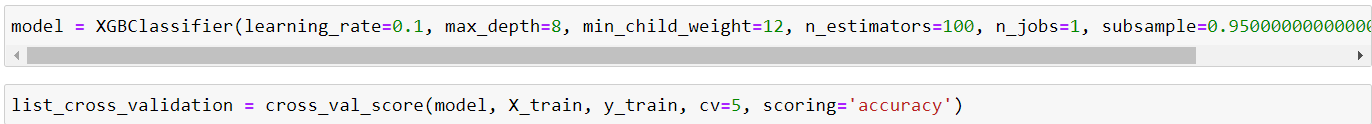
Description automatically generated

Dengan menetapkan hyperparameter max\_trials=20 dan epochs=150, didapatkan hasil akurasinya yaitu sebesar 82.27030038833618%.

Jika hanya dilihat dari hasil akurasinya, model yang dihasilkan TPOT lebih baik karena memiliki hasil akurasi yang lebih baik. Hasil akurasi pada model yang dihasilkan TPOT memiliki selisih akurasi sekitar 2.66% lebih baik jika dibandingkan dengan model yang dihasilkan oleh autokeras. Namun untuk mengetahui apakah model yang dihasilkan dari kedua tools tersebut stabil atau tidak, mengacu pada hasil akurasinya saja tidak cukup. Maka perlu dilakukan evaluasi lebih lanjut, yang akan kami jelaskan pada sub-bab selanjutnya.

* 1. Evaluasi

Untuk TPOT, metode evaluasi yang kami gunakan adalah cross validation dengan jumlah fold sebanyak 5. Model yang digunakan adalah model yang didapatkan dari hasil TPOT yaitu XGBClassifier dengan hyperparameter yang sama persis dengan hyperparameter yang ada pada hasil TPOT. Setelah dicek menggunakan cross validation, akurasi yang didapat adalah sebagai berikut:



Graphical user interface, text, application

Description automatically generated

1. Akurasi fold ke-1: 85.78168362627197%
2. Akurasi fold ke-2: 85.84643848288621%
3. Akurasi fold ke-3: 85.48566142460684%
4. Akurasi fold ke-4: 85.45656397446573%
5. Akurasi fold ke-5: 85.02174114164123%

Dengan rata-rata akurasi seluruh foldnya adalah 85.51841772997439%.

Setelah kami analisa, akurasi dari setiap fold memiliki angka yang tidak jauh berbeda yang masih berada dalam kisaran 85%. Artinya, model yang kami buat merupakan model yang stabil yang dapat diimplementasikan pada data test apapun asalkan memiliki field dan atribut yang sama dengan model kami. Perbedaan akurasi TPOT dan akurasi rata-rata cross validation juga hanya berbeda tidak lebih dari 1% yang menandakan kalau model yang kami buat tidak overfit maupun underfit.

Untuk autokeras, kami melakukan evaluasi dengan membandingkan hasil predict pada datatest dengan hasil predict pada datatrain itu sendiri. Hasilnya sebagai berikut:

Graphical user interface, text, application, email

Description automatically generated

1. Akurasi pada datatest: 82.27%
2. Akurasi pada datatrain: 84.90%

Selisih dari keduanya adalah 2.63%. Memang masih tergolong kecil untuk dikatakan overfit, tapi selisih tersebut lebih besar jika dibandingkan dengan selisih akurasi pada evaluasi model yang dihasilkan TPOT. Sehingga TPOT jelas menghasilkan model yang lebih baik dan lebih stabil.

* 1. Kesimpulan

Kesimpulan yang kami dapat dari hasil eksperimen adalah model yang dihasilkan oleh TPOT lebih baik, baik dari segi akurasi, dan kestabilan jika dibandingkan dengan model yang dihasilkan dari autokeras, sehingga TPOT akan lebih cocok digunakan untuk masalah lain yang serupa.

Sayangnya karena operating system yang tidak bisa menjalankan auto sklearn dan keterbatasan biaya karena kami tidak memiliki lisensi dari tools auto machine learning Google Cloud dan Microsoft Azure, kami tidak jadi tidak bisa untuk mengexplore lebih dalam mengenai tools-tools tersebut untuk permasalahan auto machine learning. Improvement untuk kedepannya, jika kami dapat mengakses ketiga tools tersebut, tentu saja kami sangat tertarik untuk mencobanya sehingga kami harap mungkin kami dapat menemukan tools auto machine learning yang lebih baik dari TPOT.