IF3270 PEMBELAJARAN MESIN

PRAKTIKUM

Disusun oleh

Muhammad Garebaldhie Er Rahman

13520029

M Syahrul Surya Putra

13520161

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

Daftar Isi

Hasil Analisis Data

Berikut adalah hasil analisis data dari dataset Denpasar Weather Data

a) Duplicate Value

```
duplicate = df_train.duplicated().sum()
print("Baris duplikat berjumlah: ", duplicate)

no_hour = df_train.drop("hour", axis=1).duplicated().sum()
print("Baris duplikat tanpa jam: ", no_hour)
```

```
Baris duplikat berjumlah:  3309
Baris duplikat tanpa jam:  19063
```

Didapatkan bahwa terdapat nilai duplikat berjumlah 3309 jika terdapat kolom "hour" Namun, kami mengasumsikan bahwa kolom **"hour"** ini tidak akan berpengaruh kepada nilai **"raintomorrow"**, maka kami coba untuk mendapatkan nilai duplikat ketika kolom **"hour"** di drop

b) Missing Value

```
missing = df_train.isna().sum()
print("Jumlah value yang missing: ", missing)
```

```
Jumlah value yang missing:  hour         0
temp          0
temp_min      0
temp_max      0
pressure      0
humidity      0
wind_speed    0
wind_deg      0
raining       0
dtype: int64
```

Dari dataset yang diberikan, tidak terdapat missing value

c) Outlier

```
[11]
df_outlier = df_train.select_dtypes(include=["float64", "int64"])
for x in df_outlier.columns:
    q1 = df_outlier[x].quantile(0.25)
    q3 = df_outlier[x].quantile(0.75)
    iqr = q3 - q1
    lower = q1 - 1.5 * iqr
    upper = q3 + 1.5 * iqr

    print(x, len(df_outlier[(df_outlier[x] < lower) | (df_outlier[x] > upper)]))
```

```
hour 0
temp 838
temp_min 1061
temp_max 322
pressure 682
humidity 150
wind_speed 2219
wind_deg 0
```

Didapatkan jumlah nilai yang merupakan outlier dari masing-masing kolom yang berada pada dataset. Outlier dicari dengan cara menghitung interquartile range lalu mencari data data yang keluar dari upper bound dan lower bound. Upper bound yaitu quartile ke 3 ditambah dengan 3/2 dikali intertquartile range dan lower bound yaitu quartile ke 1 dikurang dengan 3/2 dikali interquartile range.

d) Balance of Data

```
df_train["raining"].value_counts()
```

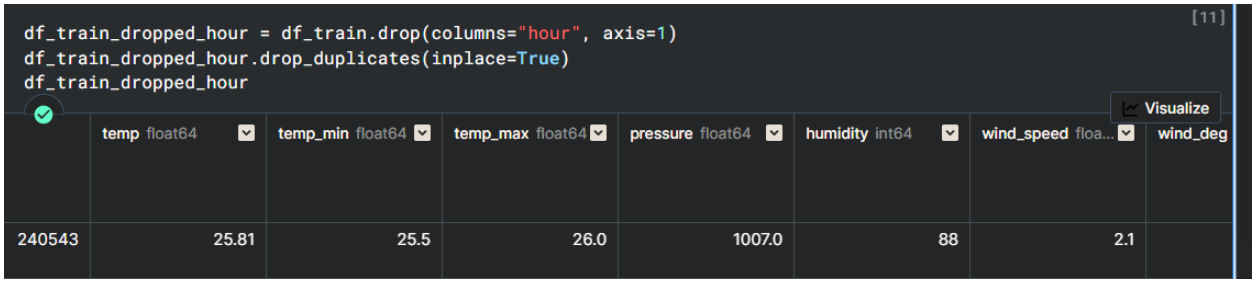
```
False    147238
True      22313
Name: raining, dtype: int64
```

Didapatkan bahwa dataset yang ada *imbalanced*. Dikarenakan proporsi nilai False dan True yang jauh

Penanganan dari Hasil Analisis Data

Berikut adalah penanganan dari hasil analisis data yang telah dilakukan:

a) Duplicate Value



```
[11]
df_train_dropped_hour = df_train.drop(columns="hour", axis=1)
df_train_dropped_hour.drop_duplicates(inplace=True)
df_train_dropped_hour
```

Visualize

	temp float64	temp_min float64	temp_max float64	pressure float64	humidity int64	wind_speed floa...	wind_deg
240543	25.81	25.5	26.0	1007.0	88	2.1	

Untuk menangani duplicate value, yang dilakukan adalah mendrop kolom “rain” terlebih dahulu, kemudian mendrop row yang memang merupakan duplikat

b) Missing Value

Dataset tidak mengandung missing value, sehingga penanganan untuk kasus ini tidak diperlukan

c) Outlier

Dari hasil dataset yang ada, diketahui terdapat beberapa nilai outlier dari masing-masing kolom. Ditanganinya adalah dengan cara mem-filter nilai yang merupakan outlier dari dataset yang ada, sehingga nanti tidak ada nilai outlier

d) Balance of Data

Untuk balance of data dilakukan oversampling dan juga undersampling. Kemudian akan diuji mana dari keduanya yang merupakan metode yang lebih baik

Justifikasi Teknik-Teknik yang Dipilih

Berikut adalah justifikasi teknik-teknik yang dipilih:

a) Duplicate Value

Kolom “hour” di drop terlebih dahulu karena diasumsikan bahwa dalam terjadinya hujan, jam tidak berpengaruh dalam nilai akhir. Selain itu, dengan dihapusnya kolom “hour”, akan menghasilkan nilai row duplikat yang lebih banyak.

Untuk row duplikat sendiri, akan dihapus untuk memperkecil overfitting dari data dan memberikan efek yang tidak diinginkan dalam model

b) Missing Value

Tidak terdalat missing value

c) Outlier

Karena terdapat beberapa data outlier maka kita cukup menghilangkannya dengan rumus yang sama. Jika sebelumnya untuk mencari outlier kita menggunakan < lower_bound atau > upper_bound maka sekarang tinggal kita balik kondisinya yaitu kita akan mencari data yang masi berada pada dalam range lower dan upper bound atau dalam kata lain lower_bound < data < upper_bound. Data ini perlu dibuang karena outlier dapat mengakibatkan hasil menjadi cukup bias

d) Balance of Data

Data yang imbalance diperlukan *oversampling* dan *undersampling* untuk membuat datanya menjadi balance. Dalam kasus sekarang, dilakukan keduanya untuk melihat performa mana yang lebih bagus kepada model dan itu merupakan *undersampling*

Perubahan dari poin 1-5

Penambahan Scaling

Setelah dilihat lebih lanjut, skala dari masing-masing kolom data yang ada bisa dibilang relatif berbeda satu sama lain. Alhasil, untuk membuat model yang dilatih lebih bagus, digunakan MinMaxScaling agar distribusi masing-masing nilai di tiap row memiliki patokan yang sama, alhasil membuat performa model lebih baik

Desain Eksperimen

a) Tujuan Eksperimen

   Eksperimen ini akan mencari hyper parameter untuk suatu model sehingga bisa memprediksi dataset dan menghasilkan model untuk mengecek apakah dengan parameter cuaca yang ada menghasilkan hujan atau tidak

b) Variabel Dependen dan Independen

   Variabel Dependen: raining

   Variabel Independen: temp, temp_min, temp_max, pressure, humidity, wind_speed, wind_deg

c) Strategi Eksperimen

   Pada eksprimen ini akan digunakan beberapa model untuk mencari prediksi rain yang paling accurate. Sebelumnya, akan dilakukan terlebih dahulu preprocessing data dengan mengacu pada bagian rencana penanganan. Kemudian akan ditambahkan penghapusan kolom hour dari data

d) Skema Validasi

   Skema validasi yang akan digunakan adalah cross-validation dengan fold 5. Umumnya terdapat k umum untuk k cross validation yaitu 1, 5 ataupun 10. Kami memililih k dengan nilai 5 karena ditujukan untuk efisiensi waktu yang bagus tetapi masih mendapatkan validasi yang akurat

Hasil Eksperimen

Setelah dilakukan preprocessing data, didapatkan bahwa dengan melakukan undersampling, akan memberikan hasil dengan waktu eksekusi yang lebih baik dari oversampling ataupun menggunakan data train yang imbalanced

Selain itu, berikut adalah metrik-metrik dari model-model mulai dari baseline hingga XGBoost

Baseline (Logistic regression)
Accuracy: 0.8728508068321224 Precision: 0.5824675324675325 Recall: 0.12830782434558718 F1: 0.21029187668503105 Confusion Matrix [[45351 643] [6094 897]]

RandomForestClassifier (Stacking DTC)
Accuracy: 0.6023025384542795 F1: 0.36365283565863377 Confusion Matrix [[25892 20102] [970 6021]] Hasil Cross Validate dengan Fold 5 Accuracy: 0.8804142805910047 Precision: 0.606150516904497 Recall: 0.27033113111196216 F1: 0.37335403311290494

XGBoost (Gradient Boosting)
Accuracy: 0.6319146928375955 F1: 0.3892142432119257 Confusion Matrix [[27268 18726] [777 6214]] Hasil Cross Validate dengan Fold 5 Accuracy: 0.8911012502559206 Precision: 0.7080725201929307 Recall: 0.2992664049534023 F1: 0.4204168479926021

Analisis dari Hasil Eksperimen

Baseline (Logistic regression)
Dari hasil training, didapatkah bahwa accuracy dari baseline, yang menggunakan logistic regression cukup tinggi. Tetapi, nilai recall dari baseline ini sangat kecil, sehingga membuat score F1 ikut kecil juga

RandomForestClassifier
Dari hasil training, didapatkah bahwa accuracy dari RFC relatif sedang. Tetapi, nilai F1 dari RFC ini lebih tinggi dari baseline. Dan ketika dilakukan cross-validate, masing-masing nilai dari accuracy dan F1 tidak memiliki penurunan

XGBoost
Dari hasil training, didapatkah bahwa accuracy dari XGBoost relatif sedang. Tetapi, nilai F1 dari XGBoost ini lebih tinggi dari baseline dan juga RFC. Dan ketika dilakukan cross-validate, masing-masing nilai dari accuracy dan F1 tidak memiliki penurunan

Kesimpulan

Jadi, setelah dilakukan training dari model-model yang kami pilih, Random Forest Classifier dan XGBoost, dengan menggunakan Logistic Regression sebagai baseline dan data yang telah kami preprocessed sesuai dengan desain eksperimen yang telah dicantumkan di atas, kami simpulkan bahwa model yang memilki prediksi yang balik baik untuk digunakan adalah XGBoost