LAPORAN PRAKTIKUM

IF3270 PEMBELAJARAN MESIN

DISUSUN OLEH

M FARREL DANENDRA RACHIM 13521048

AKMAL MAHARDIKA N P 13521070

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2024

DAFTAR ISI

BAB I

ANALISIS DATA

1.1 Hasil Analisis Data

Analisis data bertujuan bertujuan untuk memahami ciri dataset serta setiap fitur yang ada pada dataset *diabetes.csv*. Dataset tersebut telah terbagi menjadi data latih (*df_train*), data validasi (*df_val*), dan data test (*df_test*).

1.1.1. Duplicate value

Berikut adalah banyaknya baris duplikat dalam dataset ini.

```
# Duplicate value
# dataset
print("banyak baris duplikat: ", data.duplicated().sum(), "dari", data.shape[0])
print("persentase duplikat: ", data.duplicated().sum() / data.shape[0] * 100, "%")

# train
print("banyak baris duplikat: ", pd.concat(X_train, y_train).duplicated().sum(), "dari", pd.concat(X_train, y_train).
print("persentase duplikat: ", pd.concat(X_train, y_train).duplicated().sum() / pd.concat(X_train, y_train).shape[0]
```

```
banyak baris duplikat:  2329 dari 50736
persentase duplikat:  4.590428886786502 %
```

Terdapat 2329 *instance* duplikat dari 50736 baris data yang merupakan baris duplikat, dengan persentase baris yang termasuk duplikat mencapai 4.59%.

1.1.2. Missing value

Berikut adalah banyaknya baris yang memiliki *missing value* dalam dataset ini.

```
# Missing value
print("Missing values:")

missing_value = pd.DataFrame()
missing_value['jumlah missing value'] = data.isnull().sum().to_frame()
missing_value['persentase missing value (%)'] = (missing_value['jumlah missing value'] / data.shape[0]) * 100

display(missing_value)
```

```
Missing values:
```

	jumlah missing val... 0 - 0	persentase missin... 0.0 - 0.0
HighBP	0	0
HighChol	0	0
BMI	0	0
Smoker	0	0
Stroke	0	0
HeartDiseaseorAttack	0	0
PhysActivity	0	0
Fruits	0	0
Veggies	0	0
HvyAlcoholConsump	0	0
AnyHealthcare	0	0
GenHlth	0	0
MentHlth	0	0
PhysHlth	0	0
DiffWalk	0	0
Sex	0	0
Age	0	0
Education	0	0
Income	0	0
Diabetes	0	0

Tidak terdapat nilai yang hilang dalam dataset ini, jadi kasus missing value tidak perlu ditangani.

1.1.3. Outlier

Berikut adalah banyaknya baris yang memiliki *outlier* (pencilan) dalam dataset ini.

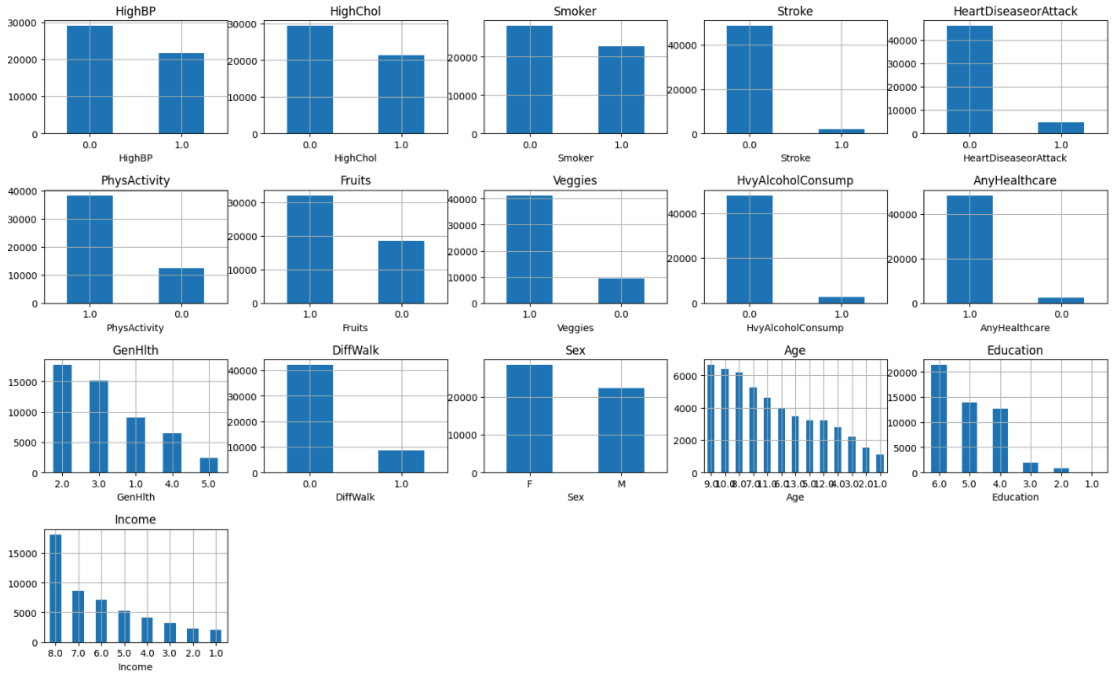
```
print("Banyak nilai outlier tiap atribut numerik:")

for col in numerical_data:
  col_data = numerical_data[col]
  iqr = col_data.quantile(.75) - col_data.quantile(.25)
  data_outlier = data.loc[(col_data < col_data.quantile(.25) - 1.5 * iqr) | (col_data > col_data.quantile(.75) + 1.5 *
  outlier_size = len(data_outlier[col])
  print(f"{col}: {outlier_size} dari {data.shape[0]}")
```

```
Banyak nilai outlier tiap atribut numerik:
BMI: 1979 dari 50736
MentHlth: 7308 dari 50736
PhysHlth: 8198 dari 50736
```

Kami menggunakan metode *interquartile range* (IQR), yaitu *outlier* data yang lebih dari upper bound (1.5* IQR + Q3) atau kurang dari lower bound (Q1 - 1.5 * IQR). Dari kolom numerik dataset ini ("BMI", "MentHlth", dan "PhysHlth"), "BMI" memiliki 1979 data *outlier*, "MentHlth" memiliki 7308, sedangkan "PhysHlth" memiliki 8198.

Selain itu, kami juga memeriksa kolom-kolom kategorikal apakah ada nilai yang tidak termasuk skala yang telah ditentukan sebelumnya menggunakan visualisasi bar chart matplotlib. Dari data tersebut, dapat dilihat bahwa tidak ada kolom yang memiliki nilai yang berbeda (sudah sesuai skala/kategori/nilai biner).



1.1.4. Balance of Data

Berikut adalah banyaknya nilai “False” dan “True” pada kolom target “Diabetes”.

```
# Balance of data
data['Diabetes'].value_counts()
```

```
Diabetes
False    43790
True      6946
Name: count, dtype: int64
```

Dari banyaknya nilai "False" dan "True" dari kelas target Diabetes di atas dapat dilihat bahwa terdapat kasus "imbalanced data" karena nilai "False" jauh lebih besar frekuensinya dibandingkan nilai "True".

1.2 Penanganan Hasil Analisis Data

1.2.1. Duplicate value

Untuk menangani nilai duplikat, kami menghilangkan nilai duplikat pada df_train memakai metode “drop_duplicates()”.

```
df_train = df_train.drop_duplicates()
```

1.2.2. Outlier

Untuk menangani *outlier*, kami menghapus baris-baris yang memiliki nilai *outlier* dari kolom-kolom numerik. Jika ada nilai yang melebihi upper bound atau kurang dari lower bound, baris data yang memiliki nilai tersebut dihapus.

```
        for col in ['BMI', 'MentHlth', 'PhysHlth']:
            col_data = numerical_data[col]
            iqr = col_data.quantile(.75) - col_data.quantile(.25)
             data_outlier_index = df_train.loc[(col_data < col_data.quantile(.25) - 1.5 * iqr) |
(col_data > col_data.quantile(.75) + 1.5 * iqr)].index
            df_train.drop(index=data_outlier_index, inplace=True)
```

1.2.3. Balance of Data

Terdapat adanya *imbalanced dataset* pada kolom target, sehingga akan dilakukan strategi oversampling pada dataset minoritas dan undersampling pada dataset mayoritas.

1.3 Justifikasi Teknik yang Dipilih

1.3.1. Duplicate value

Berdasarkan hasil analisis, ternyata sekitar 4.6% data bersifat duplikat. Hal ini tentunya akan membuat hasil pembelajaran tidak akurat. Oleh karena itu, kami akan melakukan penghapusan data duplikat agar dapat memperkecil *overfitting* dari data yang akan dilatih.

1.3.2. Outlier

Nilai *outlier* dapat mengakibatkan hasil pembelajaran menjadi cukup bias, dan akhirnya tidak seakurat yang diinginkan. Oleh karena itu, kami membuang semua data yang salah satu nilainya terletak di luar *upper* dan *lower bound* dari IQR kolom nilai tersebut.

1.3.3. Balance of Data

Dengan adanya implementasi oversampling dan undersampling, kami dapat memperoleh banyak kelas target yang proporsinya lebih setara, karena oversampling menyebabkan kelas minoritas lebih banyak, sedangkan undersampling menyebabkan kelas minoritas lebih sedikit. Untuk pembelajaran ini, sebaiknya dipilih di antara oversampling dan undersampling, yang mana yang lebih efektif untuk mempertahankan performa jalannya program.

BAB II

DESIGN OF EXPERIMENT

2.1 Desain Eksperimen

2.1.1 Tujuan Eksperimen

Eksperimen ini bertujuan untuk memprediksi fitur *Diabetes* dan menghasilkan model yang ideal untuk melakukan pembelajaran yang memeriksa apa seseorang memiliki diabetes atau tidak berdasarkan atribut-atribut independen yang ada.

2.1.2 Variabel Dependen dan Independen

Variabel dependen adalah variabel target atau label. Berdasarkan tujuan eksperimen variabel dependen adalah *Diabetes* dengan deskripsi ‘Apakah mengalami diabetes atau tidak’. Sementara itu, berikut adalah variabel independen dan deskripsinya.

Fitur/Atribut	**Deskripsi**

HighBP	Memiliki tekanan darah tinggi (BP: Blood Pressure) atau tidak
HighChol	Kolesterol tinggi atau tidak
BMI	Besaran Body Mass Index
Smoker	Perokok atau bukan perokok
Stroke	Pernah mengalami struk atau tidak
HeartDiseaseorAttack	Memiliki riwayat penyakit antara jantung koroner dan serangan jantung atau tidak sama sekali
PhysActivity	Aktif secara fisik dalam 30 hari terakhir atau tidak
Fruits	Mengonsumsi buah setiap hari atau tidak
Veggies	Mengonsumsi sayur setiap hari atau tidak
HvyAlcoholConsump	Peminum berat alkohol atau bukan
AnyHealthcare	Memiliki perlindungan kesehatan atau tidak, contohnya memiliki asuransi kesehatan
GenHtlth	Evaluasi mandiri terhadap kesehatan, skala 1-5 (1: Sangat baik, 2: Cukup Baik, 3: Baik, 4: Biasa saja, 5: Buruk)
MentHlth	Jumlah hari keadaan mental buruk dalam 30 hari terakhir (skala 0-30 hari)
PhysHlth	Jumlah hari keadaan fisik buruk dalam 30 hari terakhir (skala 0-30 hari)
DiffWalk	Memiliki kesulitan berjalan atau menaiki tangga
Sex	(M) Male atau (F) Female
Age	13 kategori umur (1: 18-24 tahun, 9: 60-64 tahun, 13: 80 tahun ke atas)
Education	Level edukasi skala 1-6 (1: Tidak pernah sekolah atau hanya TK, 2: SD, dst)
Income	Skala pendapatan 1-8

2.1.3 Strategi Eksperimen

Berikut adalah strategi tahap-tahap eksperimen yang kami laksanakan:

1. Menangani data duplikat dan outlier, serta encoding
2. Dataset split & baseline training: Membagi data menjadi data training, data validasi, dan data test, serta melakukan training awal memakai LogisticRegression. Proses ini dilakukan lagi di bagian 2 karena akan dilakukan *pre-processing* sebelum itu..

3. Hyperparameter Tuning: melakukan pengaturan hyperparameter dengan Grid Search untuk mencari kombinasi terbaik dari hyperparameter yang ada. Kami akhirnya menggunakan hyperparameter lr_params = { 'C' : [0.03, 0.04, 0.05, 1], 'max_iter' : [290, 300, 500]}
4. Melakukan training dengan model lain: Kami menggunakan model lain seperti Random Forest, KNN, dan Decision Learning Tree (DTL).
5. Melakukan oversampling pada kelas minoritas atau undersampling pada kelas mayoritas. Selain itu, model hasil tuning dilatih kembali dan dievaluasi dengan confusion matrix dan classification report.
6. Menggabungkan beberapa model yang sudah di-training di poin no. 3 dan 4 menggunakan Voting Classifier dan Stacking Classifier (terdapat dua model stacking classifier: ada yang menggabungkan logistic regression dengan DTL serta ada yang menggabungkan logistic regression, DTL, KNN, dan RandomForest (Stacking2)).
7. Memilih model final berdasarkan analisis di atas.

2.1.4 Skema Validasi

Skema validasi yang digunakan adalah 5-fold cross validation, karena dapat mempertahankan efisiensi komputasi serta memiliki estimasi yang baik, dibandingkan 10-fold yang memiliki waktu komputasi lebih lama walaupun dapat menghasilkan model yang lebih baik.

2.1.5 Perubahan dari Poin 1-5

Kami menggunakan ulang df_train, df_val, dan df_test dari inisialisasi awal sebelum bagian 1 untuk bagian 2. Sebaliknya, kami melakukan baseline training untuk kedua kalinya di awal bagian 2 memakai model LogisticRegression. Proses ini dilakukan lagi di bagian 2 karena akan dilakukan pre-processing sebelum itu terhadap df_train, df_val, dan df_test. Selain itu, KNN juga ditambahkan ke dalam pertimbangan model karena nilai F1 score yang tinggi, dan SVM tidak jadi diuji karena waktu komputasinya yang besar.

2.2 Hasil Eksperimen

2.2.1. Perbandingan Model Baseline dengan Model Lain

a. Baseline Model (pre-processed)

```
Evaluasi Logistic Regression Base Model :
            Predicted No  Predicted Yes
Actual No           6793            234
Actual Yes           850            241
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      7027
           1       0.51      0.22      0.31      1091

    accuracy                           0.87      8118
   macro avg       0.70      0.59      0.62      8118
weighted avg       0.84      0.87      0.84      8118

F1 score: 0.3077905491698595
```

b. Baseline Model (pre-processed & post-hyperparameter tuning)

```
Logistic Regression best params:  {'C': 0.04, 'max_iter': 290}
Evaluasi Logistic Regression Model setelah tuning:
            Predicted No  Predicted Yes
Actual No           6796            231
Actual Yes           857            234
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      7027
           1       0.50      0.21      0.30      1091

    accuracy                           0.87      8118
   macro avg       0.70      0.59      0.61      8118
weighted avg       0.84      0.87      0.84      8118

F1 score: 0.3007712082262211
```

c. Random Forest

```
Evaluasi Random Forest Model :
            Predicted No  Predicted Yes
Actual No           6899            128
Actual Yes           969            122
              precision    recall  f1-score   support

           0       0.88      0.98      0.93      7027
           1       0.49      0.11      0.18      1091

    accuracy                           0.86      8118
   macro avg       0.68      0.55      0.55      8118
weighted avg       0.82      0.86      0.83      8118

F1 score: 0.1819537658463833
```

d. Decision Tree Learning

```
Evaluasi Decision Tree Model :
            Predicted No  Predicted Yes
Actual No           6022           1005
Actual Yes           765            326
              precision    recall  f1-score   support

           0       0.89      0.86      0.87      7027
           1       0.24      0.30      0.27      1091

    accuracy                           0.78      8118
   macro avg       0.57      0.58      0.57      8118
weighted avg       0.80      0.78      0.79      8118

F1 score: 0.2691990090834021
```

e. KNN

```
Evaluasi KNN Model :
            Predicted No  Predicted Yes
Actual No           6843            184
Actual Yes           970            121
              precision    recall  f1-score   support

           0       0.88      0.97      0.92      7027
           1       0.40      0.11      0.17      1091

    accuracy                           0.86      8118
   macro avg       0.64      0.54      0.55      8118
weighted avg       0.81      0.86      0.82      8118

F1 score: 0.17335243553008597
```

2.2.2. Perbandingan Oversampling dan Undersampling

a. Oversampling

```
Evaluasi dari baseline model dengan oversampling:
            Predicted No  Predicted Yes
Actual No           4790           2237
Actual Yes           217            874
              precision    recall  f1-score   support

           0       0.96      0.68      0.80      7027
           1       0.28      0.80      0.42      1091

    accuracy                           0.70      8118
   macro avg       0.62      0.74      0.61      8118
weighted avg       0.87      0.70      0.74      8118

F1 score: 0.415992384578772
Cross validation score mean:  0.8875768217734855
```

b. Undersampling

```
Evaluasi dari baseline model dengan undersampling:
            Predicted No  Predicted Yes
Actual No           4672           2355
Actual Yes           208            883
              precision    recall  f1-score   support

           0       0.96      0.66      0.78      7027
           1       0.27      0.81      0.41      1091

    accuracy                           0.68      8118
   macro avg       0.62      0.74      0.60      8118
weighted avg       0.87      0.68      0.73      8118

F1 score: 0.4079464079464079
Cross validation score mean:  0.8875768217734855
```

2.2.3. Perbandingan Voting dengan Stacking

a. Hard voting

```
Evaluasi dari ensemble model dengan hard voting:
            Predicted No  Predicted Yes
Actual No           6921            106
Actual Yes           992             99
              precision    recall  f1-score   support

           0       0.87      0.98      0.93      7027
           1       0.48      0.09      0.15      1091

    accuracy                           0.86      8118
   macro avg       0.68      0.54      0.54      8118
weighted avg       0.82      0.86      0.82      8118

F1 score: 0.1527777777777778
Cross validation score mean:  0.8866110623353819
```

b. Soft voting

```
Evaluasi dari ensemble model dengan soft voting:
            Predicted No  Predicted Yes
Actual No           6059            968
Actual Yes           770            321
              precision    recall  f1-score   support

           0       0.89      0.86      0.87      7027
           1       0.25      0.29      0.27      1091

    accuracy                           0.79      8118
   macro avg       0.57      0.58      0.57      8118
weighted avg       0.80      0.79      0.79      8118

F1 score: 0.26974789915966385
Cross validation score mean:  0.8121597892888499
```

c. Stacking

```
Evaluasi dari ensemble model dengan stacking:
            Predicted No  Predicted Yes
Actual No           6729            298
Actual Yes           815            276
              precision    recall  f1-score   support

           0       0.89      0.96      0.92      7027
           1       0.48      0.25      0.33      1091

    accuracy                           0.86      8118
   macro avg       0.69      0.61      0.63      8118
weighted avg       0.84      0.86      0.84      8118

F1 score: 0.33153153153153153
Cross validation score mean:  0.8855575065847233
```

d. Stacking2

```
Evaluasi dari ensemble model dengan stacking2:
            Predicted No  Predicted Yes
Actual No           6768            259
Actual Yes           828            263
              precision    recall  f1-score   support

           0       0.89      0.96      0.93      7027
           1       0.50      0.24      0.33      1091

    accuracy                           0.87      8118
   macro avg       0.70      0.60      0.63      8118
weighted avg       0.84      0.87      0.85      8118

F1 score: 0.3261004339739616
Cross validation score mean:  0.8855575065847233
```

2.3 Analisis Hasil Eksperimen

Dari perbandingan model baseline dengan model-model lain seperti di atas, dapat disimpulkan bahwa diperoleh parameter terbaik untuk {‘C’: 0.04, ‘max_iter’: 290}. Selain itu, jika dilakukan hyperparameter tuning terhadap model baseline yang awal, hampir tidak ada perubahan dalam F1 score maupun accuracy sama sekali. Model Random Forest memiliki nilai precision, accuracy, dan F1 score yang jauh lebih kecil dibandingkan model baseline. Model Decision Tree Learning memiliki nilai accuracy yang kurang dari Random Forest namun nilai F1 score-nya lebih besar. Model KNN sebaliknya, nilai F1 score-nya adalah yang paling kecil, namun nilai accuracy-nya sangat tinggi.

Model baseline yang dilakukan oversampling memiliki nilai F1 score yang sedikit lebih besar dibandingkan model baseline yang dilakukan undersampling. Adapun rata-rata *cross validation* dan akurasinya tidak jauh berbeda.

Dalam proses menentukan model akhir, *stacking* menghasilkan nilai akurasi yang paling tinggi, dengan Stacking memiliki akurasi yang lebih kecil dari Stacking2, namun F1 score-nya lebih besar dari Stacking2. Hard voting memiliki nilai akurasi yang mirip dengan *stacking*, namun nilai F1 score-nya jauh lebih kecil. Soft voting memiliki akurasi dan F1 score yang lebih kecil dibandingkan hard voting dan stacking.

Model Stacking2 didapatkan dengan melakukan Best-Guest. Stacking2 dibuat dengan mengonfigurasi; model-model yang digabung dan mencoba menggunakan data latih asli, data undersampling, dan data oversampling.

BAB III

KESIMPULAN

Berdasarkan beberapa model dan beberapa prediksi yang telah dilakukan didapatkan Model Stracking dengan gabungan empat model, yaitu *linear regression* dengan tuning, *decision tree*, *random forest*, dan KNN adalah model terbaik. Model ini dilatih dengan data latih tanpa penanganan imbalance data. Model ini memiliki akurasi 86 % dengan *f1-score* 0.33 untuk kelas 1 dan 0.92.

BAB IV

PEMBAGIAN KERJA ANGGOTA KELOMPOK

Berikut merupakan pembagian kerja tiap anggota kelompok untuk anggota yang menulis coding. Secara keseluruhan praktikum dikerjakan dengan metode pair-programing

Tugas	**NIM**
Baseline training	13521070
Ukuran, statistik data	13521070
Duplicate value	13521070
Missing value, outlier, balance of data	13521048
Rencana penanganan, teknik encoding, desain eksperimen	13521048, 13521070
Menangani data duplikat & outlier, encoding data	13521048
Dataset split, hyperparameter tuning	13521070

Training dengan model lain	13521070
Oversampling, undersampling	13521048
Soft voting, hard voting	13521048
Stacking	13521048, 13521070
Kesimpulan	13521048, 13521070