Diabetes2
Kelompok 9

Algoritma dan Struktur Data

2208541011 - Serly Nur Rahmadhani

2208541036 - Diska Audian Maharani

2208541044 - Alifa Fitriana Putri .Y.

Exploratory Data

Disini kami menggunakan dataset diabetes2. Kami bertujuan untuk menentukan apakah berdasarkan data yang ada dalam dataset dapat ditentukan atau tidak seseorang itu merupakan penderita penyakit diabetes2 atau bukan. Diabetes2 merupakan penyakit yang sudah lumrah dijumpai dalam masyarakat. Penyakit ini merupakan penyakit yang membuat kadar gula darah meningkat akibat kelainan pada kemampuan tubuh untuk menggunakan hormon insulin.

Adapun beberapa faktor yang dapat meningkatkan resiko seseorang terkena penyakit diabetes2, yaitu:

- Faktor genetiik atau keturunan
- Berat badan yang berlebih atau obesitas
- Sering mengkonsumsi makanan atau minuman yang mengandung gula atau karbohidrat simpleks
- Kurang beraktivitas fisik atau berolahraga
- Atau memiliki kondisi tertentu, seperti tekanan darah tinggi atau hipertensi

Disini kami akan mengidentifikasi pola atau informasi yang ada dalam dataset diabetes2, yang nantinya diharapkan dapat memberikan informasi mengenai hubungan antar faktor penyebab dengan penyakit diabetes2. Sehingga, nantinya dapat ditemukan solusi untuk mengurangi resiko terkena penyakit diabetes2.

Read Data

```
data = pd.read_csv('diabetes2.csv')
```

```
data.head(10)
```

	Diabetes_binary	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAttack	PhysActivity	Fruits	...	...	AnyHealthcare	NoDocbcCost	GenHlth	MentHlth	PhysHlth	DiffWalk	Sex	Age	Education	Income
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0	0.0	0.0	...	...	1.0	0.0	5.0	18.0	15.0	1.0	0.0	9.0	4.0	3.0
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0	1.0	0.0	...	...	0.0	1.0	3.0	0.0	0.0	0.0	0.0	7.0	6.0	1.0
2	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0	0.0	1.0	...	...	1.0	1.0	5.0	30.0	30.0	1.0	0.0	9.0	4.0	8.0
3	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0	1.0	1.0	...	...	1.0	0.0	2.0	0.0	0.0	0.0	0.0	11.0	3.0	6.0
4	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0	1.0	1.0	...	...	1.0	0.0	2.0	3.0	0.0	0.0	0.0	11.0	5.0	4.0
5	0.0	1.0	1.0	1.0	25.0	1.0	0.0	0.0	1.0	1.0	...	...	1.0	0.0	2.0	0.0	2.0	0.0	1.0	10.0	6.0	8.0
6	0.0	1.0	0.0	1.0	30.0	1.0	0.0	0.0	0.0	0.0	...	...	1.0	0.0	3.0	0.0	14.0	0.0	0.0	9.0	6.0	7.0
7	0.0	1.0	1.0	1.0	25.0	1.0	0.0	0.0	1.0	0.0	...	...	1.0	0.0	3.0	0.0	0.0	1.0	0.0	11.0	4.0	4.0
8	1.0	1.0	1.0	1.0	30.0	1.0	0.0	1.0	0.0	1.0	...	...	1.0	0.0	5.0	30.0	30.0	1.0	0.0	9.0	5.0	1.0
9	0.0	0.0	0.0	1.0	24.0	0.0	0.0	0.0	0.0	0.0	...	...	1.0	0.0	2.0	0.0	0.0	0.0	1.0	8.0	4.0	3.0

10 rows × 22 columns

Bagaimana isi dari dataset?

```
In [8]: data.info()
```

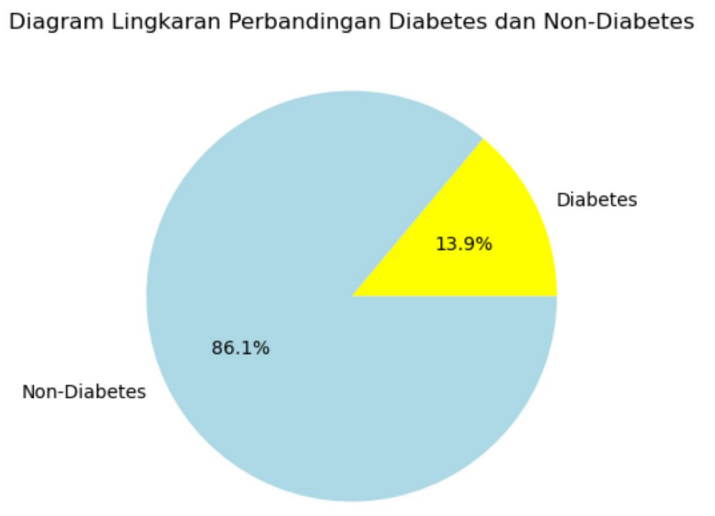
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 253680 entries, 0 to 253679
Data columns (total 22 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Diabetes_binary       253680 non-null  float64
 1   HighBP                253680 non-null  float64
 2   HighChol              253680 non-null  float64
 3   CholCheck             253680 non-null  float64
 4   BMI                   253680 non-null  float64
 5   Smoker                253680 non-null  float64
 6   Stroke                253680 non-null  float64
 7   HeartDiseaseorAttack  253680 non-null  float64
 8   PhysActivity          253680 non-null  float64
 9   Fruits                253680 non-null  float64
 10  Veggies               253680 non-null  float64
 11  HvyAlcoholConsump     253680 non-null  float64
 12  AnyHealthcare         253680 non-null  float64
 13  NoDocbcCost           253680 non-null  float64
 14  GenHlth               253680 non-null  float64
 15  MentHlth              253680 non-null  float64
 16  PhysHlth              253680 non-null  float64
 17  DiffWalk              253680 non-null  float64
 18  Sex                   253680 non-null  float64
 19  Age                   253680 non-null  float64
 20  Education             253680 non-null  float64
 21  Income                253680 non-null  float64
dtypes: float64(22)
memory usage: 42.6 MB
```

Dataset diabetes2 berukuran (253680, 22), dimana tidak terdapat mising value (data kosong) dan juga data berbentuk numerik serta bertipe data float.

Diagram Diabetes_binary

Diabetes Values
35346

Non-Diabetes Values
218334



Unique Values
[0, 1]

Data Target
Diabetes_binary

Di atas merupakan diagram mengenai data target, yaitu Diabetes_binary.

Korelasi

```
In [14]: correlations = data.corr()

In [15]: print(correlations["Diabetes_binary"])
```

```
Diabetes_binary          1.000000
HighBP                   0.263129
HighChol                 0.200276
CholCheck                0.064761
BMI                      0.216843
Smoker                   0.060789
Stroke                   0.105816
HeartDiseaseorAttack     0.177282
PhysActivity            -0.118133
Fruits                  -0.040779
Veggies                 -0.056584
HvyAlcoholConsump       -0.057056
AnyHealthcare            0.016255
NoDocbcCost              0.031433
GenHlth                  0.293569
MentHlth                 0.069315
PhysHlth                 0.171337
DiffWalk                 0.218344
Sex                      0.031430
Age                      0.177442
Education               -0.124456
Income                  -0.163919
Name: Diabetes_binary, dtype: float64
```

5 Variabel dengan korelasi tertinggi

Dari hasil korelasi dengan data target, kami mengambil 5 variabel yang memiliki korelasi paling besar dengan data target, yaitu HighBP, HighChol, BMI, GenHlth, dan DiffWalk. Lalu, variabel yang kurang memiliki pengaruh dengan data target akan didrop sehingga dapat memudahkan dilakukannya tahap selanjutnya.

Dataset Setelah Drop Columns

```
In [17]: data.head(10)
```

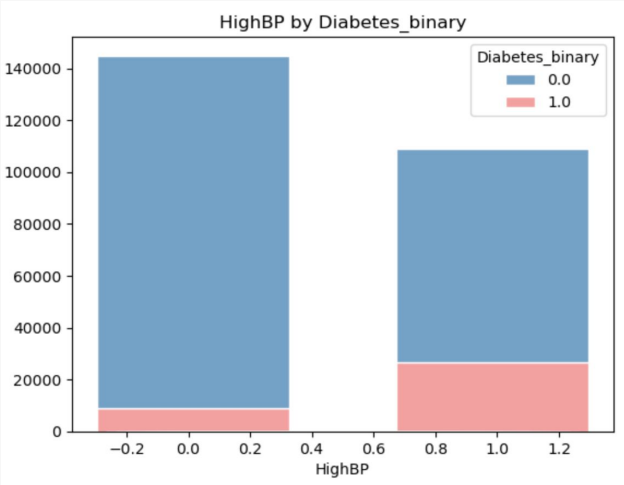
Out[17]:

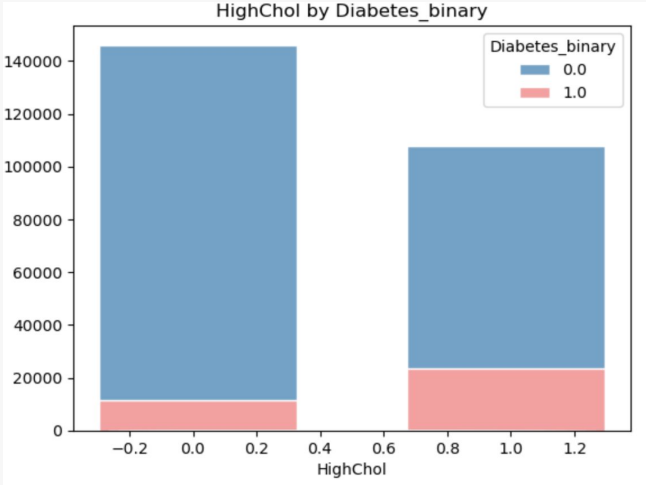
	Diabetes_binary	HighBP	HighChol	BMI	GenHlth	DiffWalk
0	0.0	1.0	1.0	40.0	5.0	1.0
1	0.0	0.0	0.0	25.0	3.0	0.0
2	0.0	1.0	1.0	28.0	5.0	1.0
3	0.0	1.0	0.0	27.0	2.0	0.0
4	0.0	1.0	1.0	24.0	2.0	0.0
5	0.0	1.0	1.0	25.0	2.0	0.0
6	0.0	1.0	0.0	30.0	3.0	0.0
7	0.0	1.0	1.0	25.0	3.0	1.0
8	1.0	1.0	1.0	30.0	5.0	1.0
9	0.0	0.0	0.0	24.0	2.0	0.0

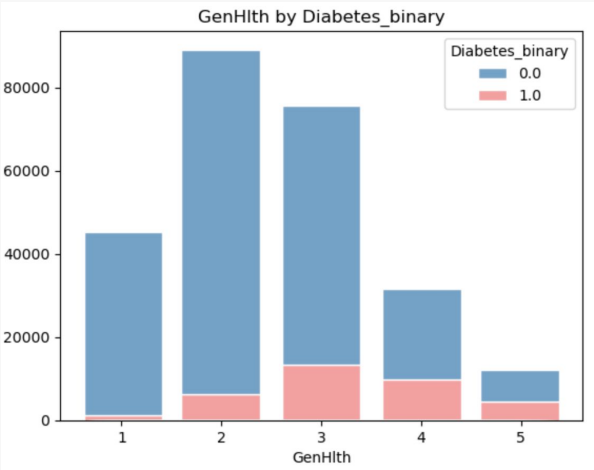
Dataset: Diabetes2

Sehingga, setelah dilakukan drop kolom terhadap variabel yang kurang berpengaruh dengan data target, dataset diabetes2 menjadi berukuran (253680, 6).

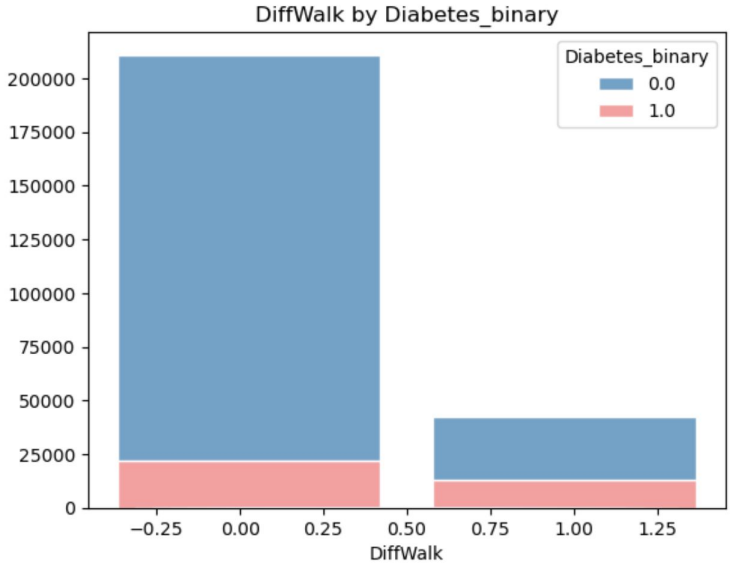
Data Understanding

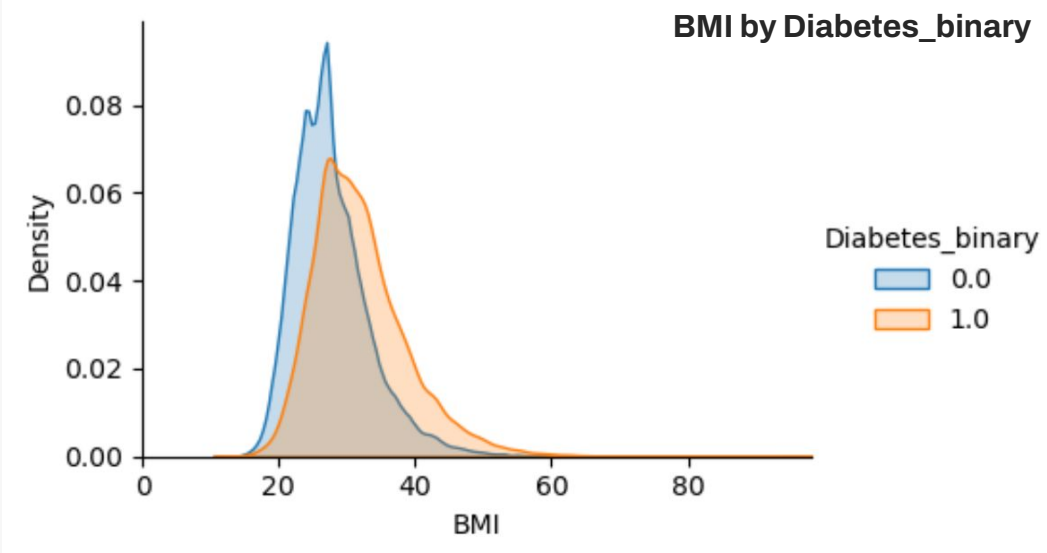






Data Understanding





Normalisasi Data

Normalisasi Data

```
n [26]: from sklearn import preprocessing

        scaler = preprocessing.MinMaxScaler()
        d = scaler.fit_transform(data)
        data_normalisasi = pd.DataFrame(d, columns = data.columns)
        data_normalisasi.head(10)
```

ut[26]:

	Diabetes_binary	HighBP	HighChol	BMI	GenHlth	DiffWalk
0	0.0	1.0	1.0	0.325581	1.00	1.0
1	0.0	0.0	0.0	0.151163	0.50	0.0
2	0.0	1.0	1.0	0.186047	1.00	1.0
3	0.0	1.0	0.0	0.174419	0.25	0.0
4	0.0	1.0	1.0	0.139535	0.25	0.0
5	0.0	1.0	1.0	0.151163	0.25	0.0
6	0.0	1.0	0.0	0.209302	0.50	0.0
7	0.0	1.0	1.0	0.151163	0.50	1.0
8	1.0	1.0	1.0	0.209302	1.00	1.0
9	0.0	0.0	0.0	0.139535	0.25	0.0

Normalisasi data digunakan untuk mengubah data asli menjadi bentuk yang lebih efisien dan memiliki struktur yang bagus. Dimana data berada dalam rentang yang lebih kecil, seperti -1 hingga 1 atau 0 hingga 1. Sehingga data yang sudah di normalisasi memungkinkan untuk dianalisis dengan lebih mudah menggunakan metode tertentu.

Split dan Test Data

Menentukan data predictor dan data target

```
]: X = data_normalisasi.drop(["Diabetes_binary"], axis = 1)
   y = data_normalisasi["Diabetes_binary"]
```

Melakukan Split dan Test Data

```
1]: from sklearn.model_selection import train_test_split

2]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
```

Karena data target imbalance, maka akan dilakukan balancing data sehingga data nya seimbang dan tidak terjadi underfitting.

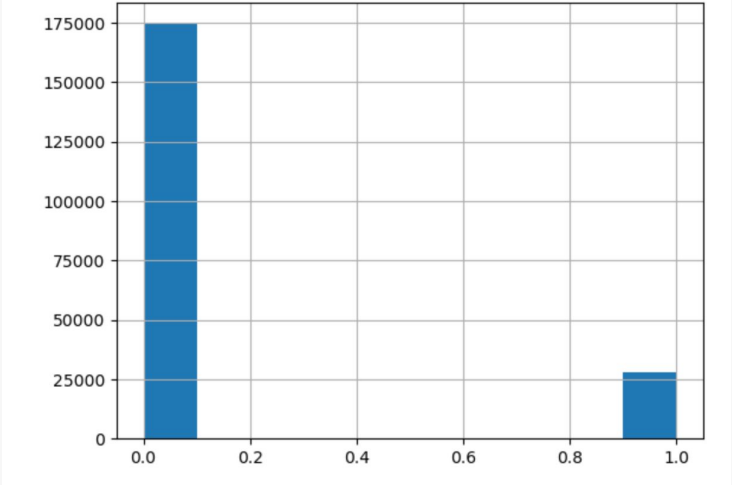
Balancing Data

Balancing Data menggunakan Random Over Sampler

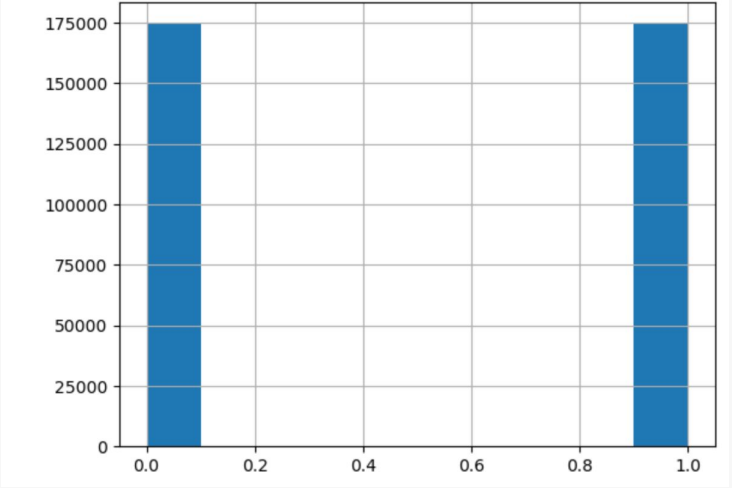
```
34]: from imblearn.over_sampling import RandomOverSampler

35]: X_os, y_os = RandomOverSampler().fit_resample(X_train, y_train)
```

Setelah dilakukan balancing data, akan terlihat bahwa data sudah seimbang.

```
]: y_train.hist()
]: <AxesSubplot:>
```




```
]: y_os.hist()
]: <AxesSubplot:>
```



Predictive Modeling

Logistic Regression

Logistic Regression adalah metode statiska yang digunakan untuk memodelkan hubungan antara variabel independen (biasanya dalam bentuk kategori atau biner) dengan variabel dependen yang juga dalam bentuk biner.

```
from sklearn.linear_model import LogisticRegression
```

```
model_lr = LogisticRegression()
```

```
##Dengan Over Sampling
model_lr.fit(X_os, y_os)
```

```
LogisticRegression()
```

```
model_lr.score(X_test, y_test)
```

```
0.7246136865342163
```

```
##Tanpa Over Sampling
model_lr.fit(X_train, y_train)
```

```
LogisticRegression()
```

```
model_lr.score(X_test, y_test)
```

```
0.8610059918006938
```

KFold Logistic Regression

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
k_fold = KFold(n_splits = 10, shuffle = True, random_state = 0)
```

```
score_lr = cross_val_score(model_lr, X, y, cv = k_fold, n_jobs = 1, scoring = 'accuracy')
print(score_lr)
```

```
[0.86163671 0.86005992 0.86431725 0.86297698 0.86143961 0.86151845
 0.86782561 0.8624251  0.86565752 0.86609114]
```

```
score_lr.mean() * 100
```

```
86.33948281299276
```

Predictive Modeling

K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah sebuah metode yang digunakan untuk mengklasifikasikan suatu objek berdasarkan kemiripin antara data baru dengan sejumlah data pada lokasi terdekat yang telah tersedia.

```
from sklearn.neighbors import KNeighborsClassifier
```

```
##Dengan Over Sampling
model_knn = KNeighborsClassifier(n_neighbors = 3)
model_knn.fit(X_os, y_os)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
model_knn.score(X_test, y_test)
```

```
0.8355211289813939
```

```
##Tanpa Over Sampling
model_knn = KNeighborsClassifier(n_neighbors = 3)
model_knn.fit(X_train, y_train)
```

```
KNeighborsClassifier(n_neighbors=3)
```

```
model_knn.score(X_test, y_test)
```

```
0.8323281299274676
```

KFold KNN

```
score_knn = cross_val_score(model_knn, X, y, cv = k_fold, n_jobs = 1, scoring = 'accuracy')
print(score_knn)
```

```
[0.82824819 0.82895774 0.82414853 0.82895774 0.82671082 0.81894513
 0.82749921 0.82694734 0.82611952 0.83199306]
```

```
score_knn.mean() * 100
```

```
82.68527278461055
```

Kesimpulan

Dari metode-metode yang kami gunakan, motode Logistic Regression merupakan metode yang memiliki tingkat akurasi paling tinggi diantara metode yang lainnya, yaitu sebesar 86.33%. Sehingga, dapat dikatakan bahwa metode Logistic Regression dapat mengklasifikasikan kasus diabetes2 ini dengan baik.

Selain itu, diperoleh juga solusi untuk mengurangi resiko terkena penyakit diabetes2. Adapun solusinya, yaitu:

- Menghindari hal-hal yang menyebabkan kadar kolestrol meningkat
- Menghindari makanan atau minuman yang menyebabkan kadar gula darah meningkat dan menyebabkan obesitas
- Menghindari hal-hal yang menyebabkan tekanan darah tinggi meningkat
- Melakukan lebih banyak aktivitas fisik atau berolahraga

Namun, terdapat juga kondisi yang tidak dapat dihindari yaitu adanya faktor genetik atau keturunan yang kemungkinan bisa menyebabkan terkena penyakit diabetes2.