Diabetes
Classification & Regression

Alyza Rahima Pramudya – 2502032125
Shafa Amira Qonitatin – 2502009173

Notes: This project was done in groups but I created this Slide Presentation myself.

Data Description

Pada project ini kami akan melakukan **klasifikasi dan regresi** menggunakan data diabetes yang dapat diakses melalui link berikut ini : **[diabetes.csv]**

Dataset ini terdiri dari beberapa feature terkait dengan kondisi seseorang yang memiliki atau tidak memiliki diabetes. Untuk memahami lebih lanjut terkait feature-feature yang ada, berikut merupakan penjelasan untuk tiap featurenya:

- **Pregnancies:** Jumlah Kehamilan
- **Glucose:** tingkat Glukosa dalam darah
- **BloodPressure:** ukuran tekanan darah
- **SkinThickness:** ukuran ketebalan kulit
- **Insulin:** ingkat Insulin dalam darah
- **BMI:** ukuran indeks massa tubuh
- **DiabetesPedigreeFunction:** persentase Diabetes
- **Age:** umur
- **Outcome:** diabetes (1), tidak diabetes (0)

Karena pada project ini kita akan melakukan klasifikasi dan regresi, maka kolom targetnya da dua. Kolom target untuk klasifikasi adalah **outcome** sedangkan untuk regresi adalah **DiabetesPedigreeFunction**

Exploratory Data Analysis (EDA)

```
df.shape
```

```
(768, 9)
```

Dataset kami terdiri dari 768 entri/pengamatan dan 9 kolom. Hal ini menunjukkan bahwa jumlah dataset yang kami miliki sudah cukup untuk melakukan eksperimen ini namun masih tergolong cukup kecil.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       752 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

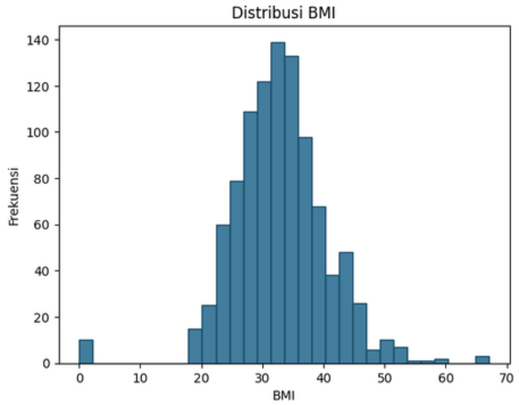
Kita dapat melihat bahwa tipe data bervariasi antara int64 dan float64. Terlihat bahwa ada juga **missing values (NaN) di kolom BMI**.

```
df.isnull().sum()
```

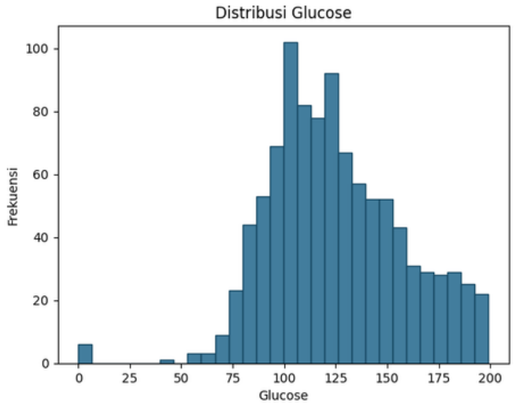
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                        16
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Setelah kita cek, ternyata ada 16 data pada kolom BMI yang merupakan missing values. Hal ini berarti 2.12% data kita merupakan missing values

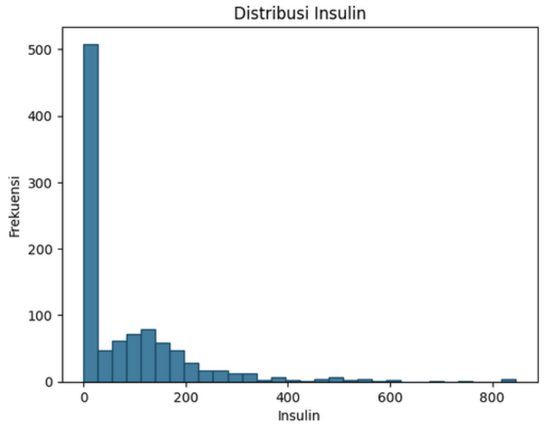
Exploratory Data Analysis (EDA)



Mostly BMI-nya berada di sekitar 30-40. Namun, ada beberapa juga yang BMI-nya diatas 50 hal ini menandakan bahwa ada beberapa orang dalam dataset yang mengalami obesitas berat.



Kadar glukosa darah pada orang-orang dalam dataset ini cukup bervariasi. Sebagian besar orang memiliki kadar glukosa darah yang terkendali, tetapi ada juga beberapa orang yang memiliki kadar glukosa darah yang tinggi.



Tingkat insulin pada orang-orang dalam dataset ini juga cukup bervariasi, dengan sebagian besar orang memiliki tingkat insulin yang rendah. Namun, ada juga beberapa orang yang memiliki tingkat insulin yang tinggi. Tingkat insulin yang rendah dapat mengindikasikan diabetes tipe 2

Exploratory Data Analysis (EDA)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1.000000	0.105628	0.157549	-0.057544	-0.062178	0.010779	-0.013678	0.531538	0.219915
Glucose	0.105628	1.000000	0.107317	0.032004	0.322887	0.233092	0.135538	0.248386	0.465286
BloodPressure	0.157549	0.107317	1.000000	0.229819	0.091322	0.246565	0.050180	0.246053	0.047670
SkinThickness	-0.057544	0.032004	0.229819	1.000000	0.424811	0.351690	0.185606	-0.091600	0.062943
Insulin	-0.062178	0.322887	0.091322	0.424811	1.000000	0.120405	0.139755	0.029334	0.124105
BMI	0.010779	0.233092	0.246565	0.351690	0.120405	1.000000	0.116683	0.020557	0.318035
DiabetesPedigreeFunction	-0.013678	0.135538	0.050180	0.185606	0.139755	0.116683	1.000000	0.044442	0.167874
Age	0.531538	0.248386	0.246053	-0.091600	0.029334	0.020557	0.044442	1.000000	0.251590
Outcome	0.219915	0.465286	0.047670	0.062943	0.124105	0.318035	0.167874	0.251590	1.000000

Berdasarkan heat map diatas, kita bisa lihat bahwa feature-feature yang ada di data kita semuanya memiliki korelasi yang positif terhadap feature Outcome. Namun terhadap feature DiabetesPedigreeFunction ada 1 feature yaitu pregnancies yang berkorelasi negatif dengan Diabetes pedigree function, itupun korelasinya sangat rendah hanya sekitar -0.01 saja.

Secara singkat, feature yang korelasinya paling tinggi dengan feature Outcome adalah feature Glucose, sedangkan feature yang korelasinya paling tinggi dengan feature DiabetesPedigreeFunction adalah feature SkinThickness.

Feature Engineering

A) Handle Null Values

BEFORE

```
df.isnull().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                        16
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

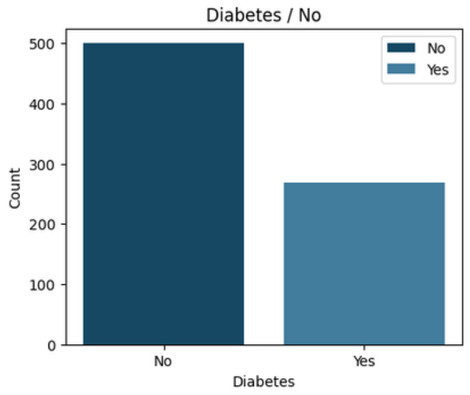
AFTER

```
df.isnull().sum()
```

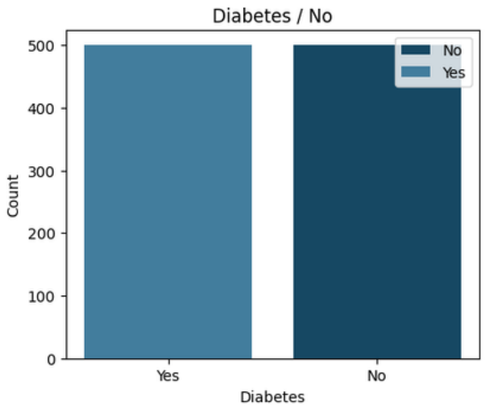
```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Pada project kali ini, walau data missing valuenya hanya sedikit, kita tidak akan drop missing value tapi kita akan **impute missing valuenya menggunakan nilai mean dari kolom tersebut** karena data yang kita miliki uga tidak banyak.

B) Handle Imbalance Data



BEFORE



AFTER

Seperti yang kita lihat, jumlah data untuk masing-masing kelas itu tidak seimbang. Maka dari itu kita akan lakukan **oversampling agar datanya menjadi seimbang dan model tidak akan bias kesalah satu kelas saja.**

Feature Engineering

C)Train Test Split

Ditahap ini kita akan split datanya **80% untuk train** dan **20% untuk test**. Data test ini akan digunakan untuk mengevaluasi kinerja model kita. Karena pada project ini kita akan melakukan **klasifikasi** dan **regresi** maka dari itu kita akan buat 2 split yang berbeda. Split untuk klasifikasi yang menjadi y atau targetnya adalah kolom **outcome** sedangkan untuk regresi yang menjadi y atau targetnya adalah kolom **diabetesPedigreeFunction.**

REGRESSION

```
X = df.drop('DiabetesPedigreeFunction', axis=1)
y = df['DiabetesPedigreeFunction']
```

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

CLASSIFICATION

```
X = df.drop(['Outcome', 'Pregnancies', 'SkinThickness', 'Insulin', 'BloodPressure'], axis=1)
y = df['Outcome']
```

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

D) Standarization

Ditahap ini kita akan standarisasi datanya menggunakn **standardscaler** agar data yang kita miliki mempunyai scala yang sama sehingga tidak akan ada feature yang terlalu mendominasi.

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Model Performances

Classification

Decision Tree

	precision	recall	f1-score	support
0	0.90	0.77	0.83	105
1	0.78	0.91	0.84	95
accuracy			0.83	200
macro avg	0.84	0.84	0.83	200
weighted avg	0.84	0.83	0.83	200

KNN

	precision	recall	f1-score	support
0	0.76	0.67	0.71	105
1	0.68	0.77	0.72	95
accuracy			0.71	200
macro avg	0.72	0.72	0.71	200
weighted avg	0.72	0.71	0.71	200

Logistic Regression

	precision	recall	f1-score	support
0	0.83	0.77	0.80	105
1	0.76	0.82	0.79	95
accuracy			0.80	200
macro avg	0.80	0.80	0.79	200
weighted avg	0.80	0.80	0.80	200

Naive Baiyes

	precision	recall	f1-score	support
0	0.80	0.82	0.81	105
1	0.79	0.77	0.78	95
accuracy			0.80	200
macro avg	0.79	0.79	0.79	200
weighted avg	0.79	0.80	0.79	200

Model Performances

Regression

Decision Tree

Mean Squared Error (Decision Tree): 0.13371916

KNN

Mean Squared Error (KNN): 0.08476422722222222

Linear Regression

Mean Squared Error (Linear Regression): 0.080118055890170 44

Kesimpulan

Pada **classification-task** yang sudah dilakukan dapat disimpulkan bahwa **model yang memiliki performances terbaik adalah model Decision Tree** diikuti dengan nilai accuracy yang lebih tinggi dibandingkan model yang lain (0.83), dimana dataset yang dimiliki adalah balanced maka dengan melihat accuracy saja sudah dapat mewakiliki performances keseluruhan model.

	precision	recall	f1-score	support
0	0.90	0.77	0.83	105
1	0.78	0.91	0.84	95
accuracy			0.83	200
macro avg	0.84	0.84	0.83	200
weighted avg	0.84	0.83	0.83	200

Pada **regression-task** yang sudah dilakukan dapat disimpulkan bahwa **model yang memiliki performances terbaik adalah model Linear Regression,** dimana MSE yang diperoleh berkisar pada 0.080. Ini menunjukkan bahwa model Linear Regression yang telah dibuat memiliki kemampuan yang baik dalam memprediksi nilai Diabetes Pedigree Function (DPF). Semakin rendah Mean Squared Error (MSE) dari model tersebut, maka semakin akurat model dalam memprediksi nilai DPF, dimana hasil prediksinya memiliki selisih yang kecil dengan nilai sebenarnya.

```
Mean Squared Error (Linear Regression): 0.08011805589017044
```

Attachment

Untuk mengakses full code dari project ini, kindly click logo google drive berikut:

BINUS UNIVERSITY

THANK YOU!