Metamorfosa Team

DigitalSkola

Diabetes Prediction

Early prediction to reduce the effects of diabetes

Our team,

Vanindra
Widya Syafira
Rendi Septian
Febri M Sitompul
Erzal Fahrezy

Penyakit Diabetes

Diabetes, adalah suatu gangguan metabolisme di mana kadar gula darah tinggi dalam waktu lama.

1. Kelebihan berat badan yang mengarah pada nilai BMI yang diatas normal
2. Tinggi angka glukosa (gula darah) dalam tubuh yang melebihi batas normal
3. Kekurangan insulin atau berkurangnya efektivitas insulin. Dimana insulin berfungsi menurunkan kadar gula dalam darah.
4. Usia > 40 tahun yang memperbesar resiko terkena diabetes.
5. Ada riwayat keluarga yang terkena diabetes.

** Tekanan darah yang tinggi (hipertensi). Hipertensi atau tekanan darah tinggi bisa terjadi dikarenakan adanya komplikasi penyakit diabetes yang kronis

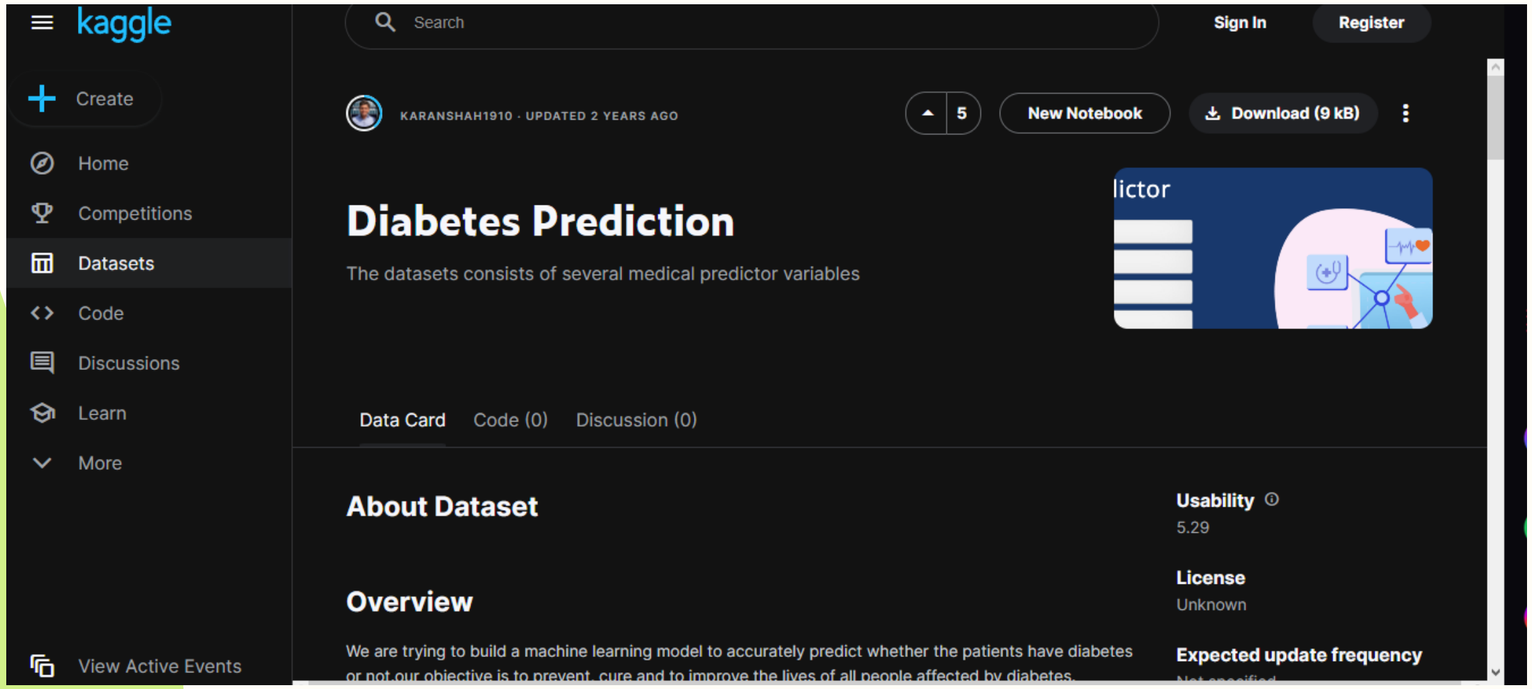
Learning Agenda

DigitalSkola

What we'll discuss this afternoon

- Goal of the Project
- Data Profiling
- Profiling Descriptive Statistics
- Data Preprocessing
- Explatory Data Analysis (EDA)
- Feature Engineering
- Feature Scaling
- Balancing Data (Oversampling)
- Modelling & Evaluation
- Hyperparameter Tuning

Sumber Data



Dataset Diabetes Prediction diambil dari website Kaggle

https://www.kaggle.com/datasets/karanshah1910/diabetes-prediction

DigitalSkola

GOAL OF THE PROJECT

Pada proyek ini, diberikan sebuah dataset kesehatan berisikan hasil tes diabetes. Diabetes sendiri merupakan salah satu penyakit berbahaya yang sampai sekarang belum ada obat. Selain itu, kasus diabetes juga sering dideteksi terlambat, sehingga penanganan yang tepat seringkali gagal diberikan.

Project ini bertujuan untuk membuat sebuah model machine learning yang dapat memprediksi apakah seseorang memiliki diabetes atau tidak. Project ini kemudian diharapkan dapat membantu dalam mendeteksi diabetes pada seseorang lebih dini sehingga ia dapat diberikan penanganan yang sesuai dan efektif.

DATA PROFILING

N = Numerical
C = Categorical

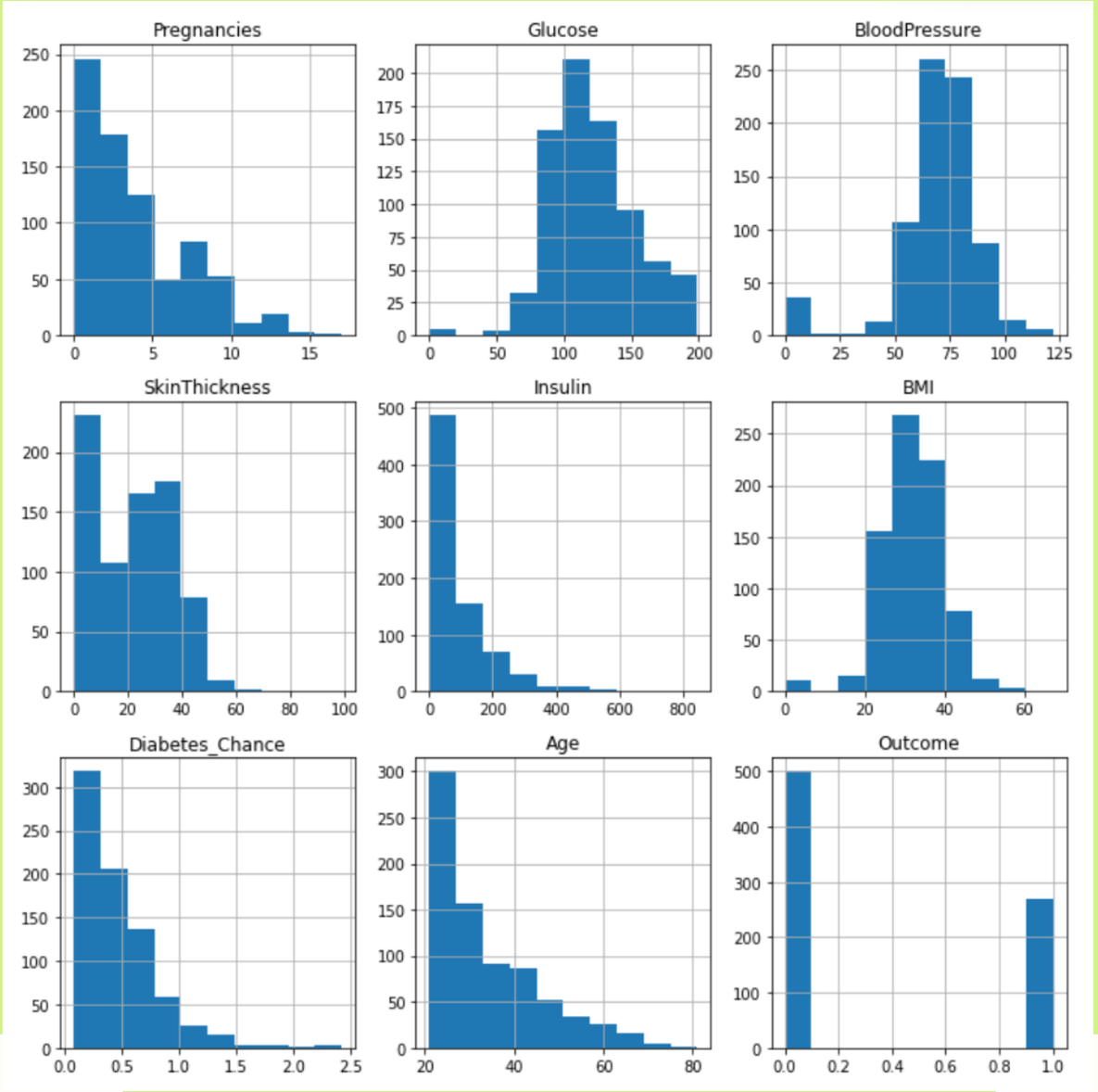
Pregnancies (N)	Jumlah kehamilan
Glucose (N)	Konsentrasi glukosa plasma pada 2 jam dalam tes toleransi glukosa
BloodPressure (N)	Tekanan darah dalam satuan mm Hg.
SkinThickness (N)	Ketebalan kulit pada bagian trisep ukuran lemak tubuh
Insulin (N)	Tingkat insulin 2 jam insulin serum
BMI (N)	Indeks Massa Tubuh (berat badan/tinggi badan ^2)
Diabetes Chance (N)	Indikator riwayat diabetes dalam keluarga
Age (N)	Usia pasien
Outcome (C)	Class variable (0 or 1), 0 untuk tidak mengidap diabetes, dan 1 mengidap diabetes.

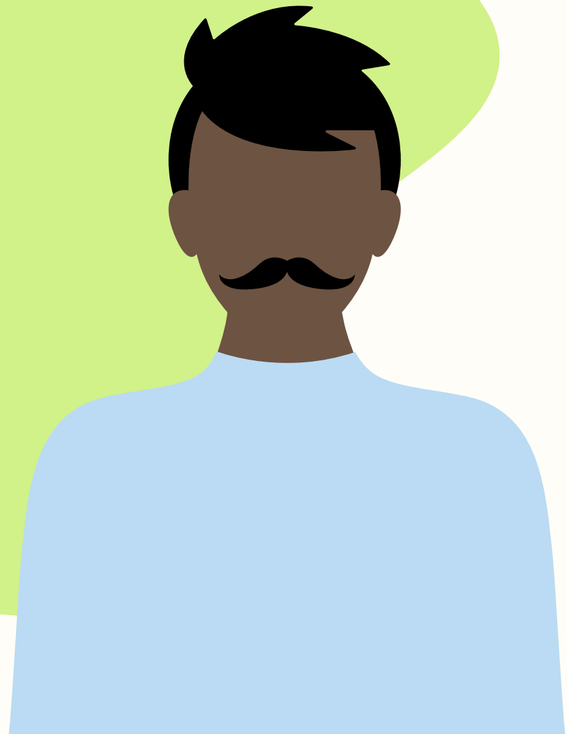
PROFILING DESCRIPTIVE STATISTICS

DigitalSkola

```
data.shape
```

```
(768, 9)
```



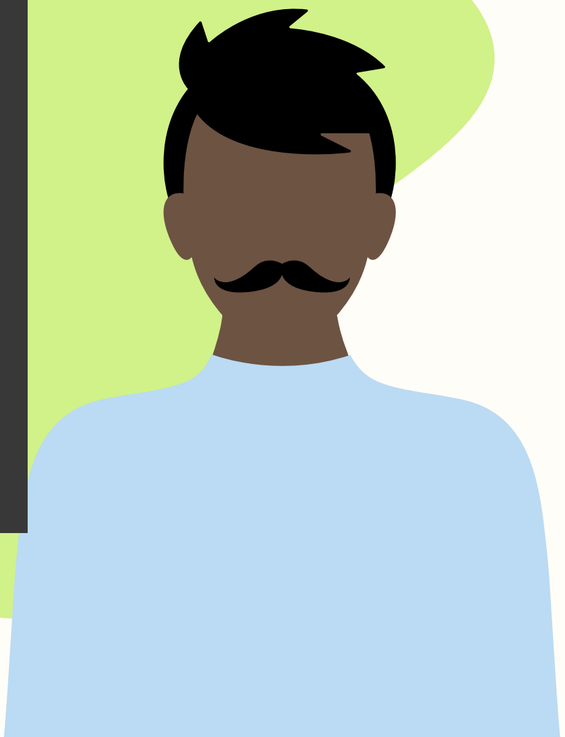


PROFILING DESCRIPTIVE STATISTICS

DigitalSkola

```
[ ]  data.info()
```

```
<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               768 non-null    float64
 6   Diabetes_Chance   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
```

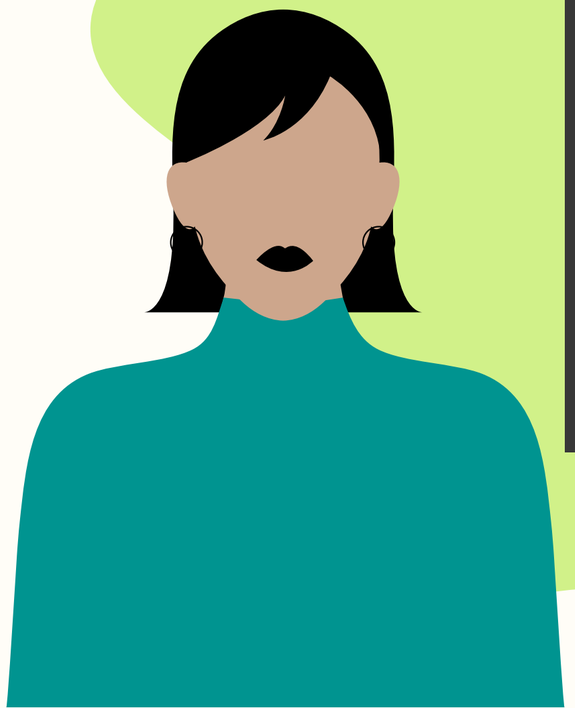


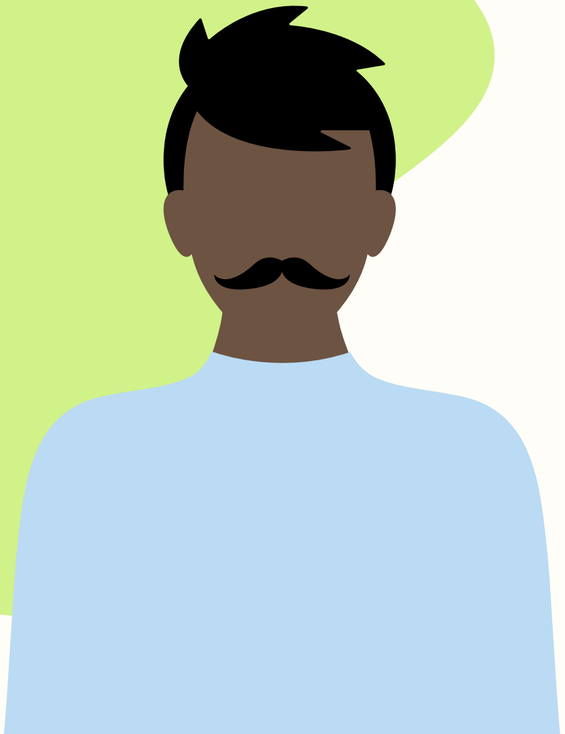
PROFILING DESCRIPTIVE STATISTICS

DigitalSkola

```
[ ]  data.isnull().sum()

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





PROFILING DESCRIPTIVE STATISTICS

DigitalSkola

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes_Chance	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Cukup aneh karena mustahil nilai Glucose, BloodPressure, SkinThickness, Insulin, BMI seseorang bernilai 0 (nol)

PROFILING DESCRIPTIVE STATISTICS

DigitalSkola

```
[ ]  data['BMI']=data['BMI'].replace([0],data['BMI'].median())
     data['Glucose']=data['Glucose'].replace([0],data['Glucose'].median())
     data['BloodPressure']=data['BloodPressure'].replace([0],data['BloodPressure'].median())
     data['SkinThickness']=data['SkinThickness'].replace([0],data['SkinThickness'].median())
     data['Insulin']=data['Insulin'].replace([0],data['SkinThickness'].median())
     data.describe()
```

PROFILING DESCRIPTIVE STATISTICS

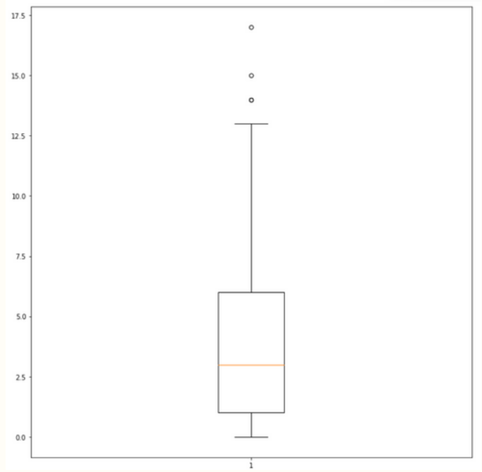
DigitalSkola

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes_Chance	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.656250	72.386719	27.334635	91.000000	32.450911	0.471876	33.240885	0.348958
std	3.369578	30.438286	12.096642	9.229014	107.812757	6.875366	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	23.000000	23.000000	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

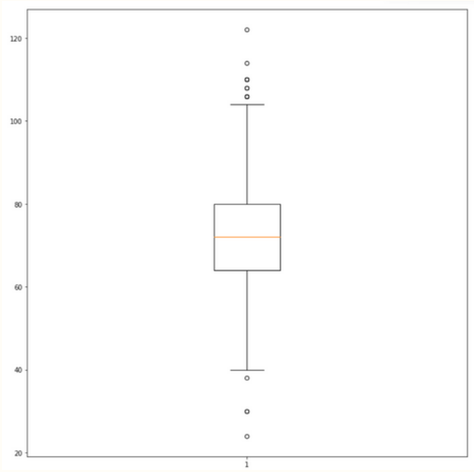
Mengganti nilai 0 (nol) pada Glucose, BloodPressure, SkinThickness, Insulin, dan BMI tiap pasien menjadi median dari masing-masing kolom.

DATA PREPROCESSING

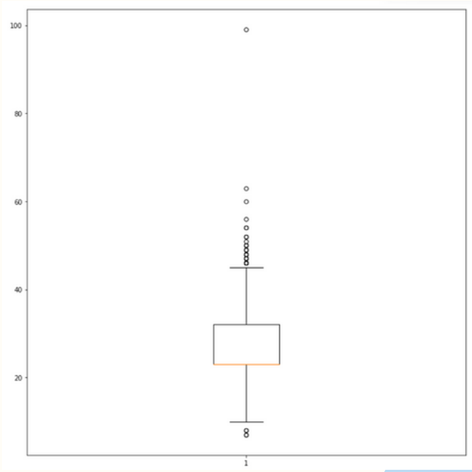
Handling Outlier



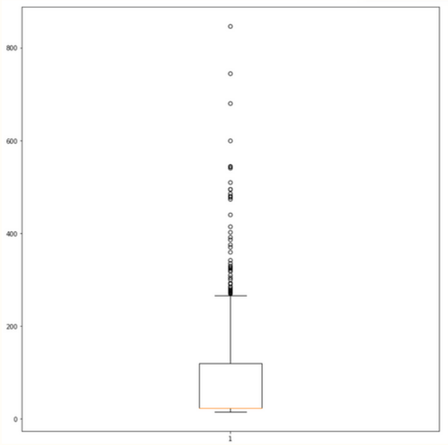
Pregnancies



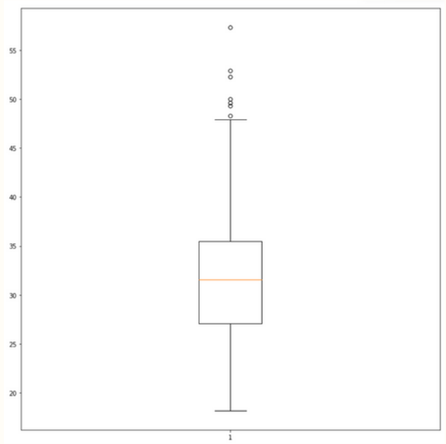
BloodPressure



SkinThickness



Insulin



BMI

Outlier yang ada tidak kami drop karena dianggap dapat menghilangkan informasi penting dari data kesehatan pasien. Outlier ini nantinya akan dijadikan kategori baru pada saat feature engineering.

```
data.shape
```

```
(768, 9)
```

Pregnancies	4 row
BloodPressure	14 row
SkinThickness	33 row
Insulin	49 row
BMI	7 row
TOTAL OUTLIER	**107 row**

DATA PREPROCESSING

Handling Duplicated Values

```
data[data.duplicated()] #There are no duplicated values
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diabetes_Chance	Age	Outcome

Tidak ditemukan duplicated values pada dataset

Checking Imbalanced Data

```
# Ketimpangan data
data.groupby('Outcome').size()
```

```
Outcome
0    500
1    268
dtype: int64
```

```
data.Outcome.value_counts(normalize=True)
```

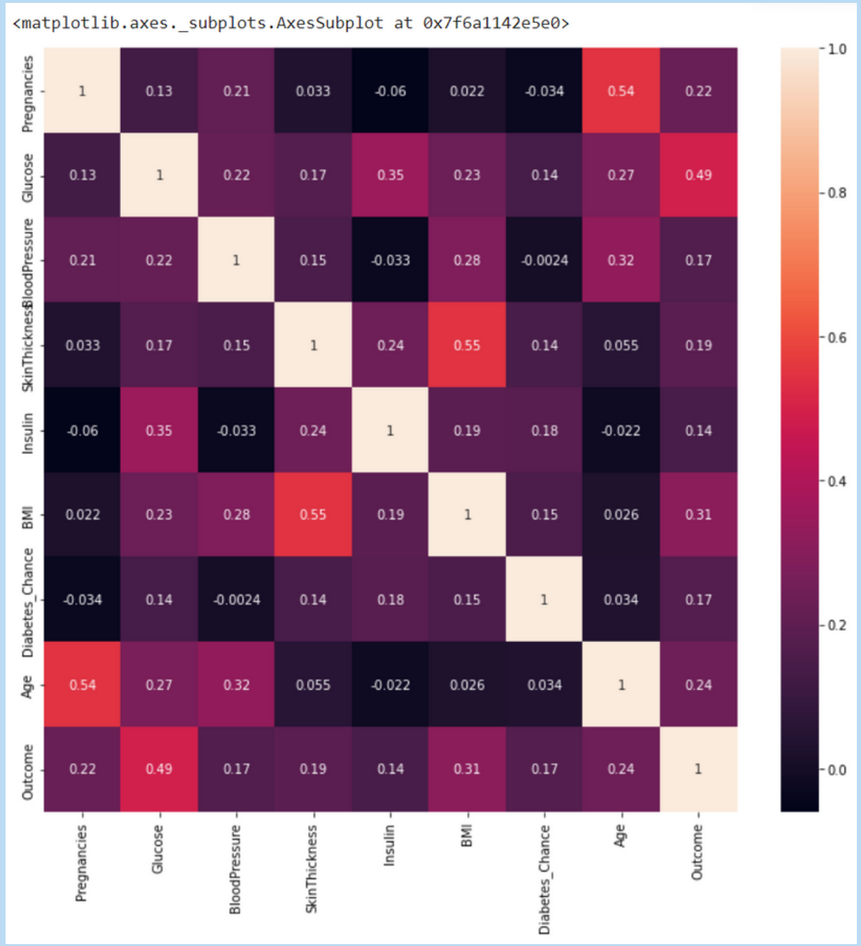
```
0    0.651042
1    0.348958
Name: Outcome, dtype: float64
```

Data yang diberikan **timpang antara outcome 0 dan 1**. Jumlah yang **tidak terkena diabetes hampir 2x jumlah yang terkena diabetes**. Ini menunjukkan perlunya balancing data dengan oversampling atau undersampling.

EXPLATORY DATA ANALYSIS (EDA)

DigitalSkola

Correlation Between Features

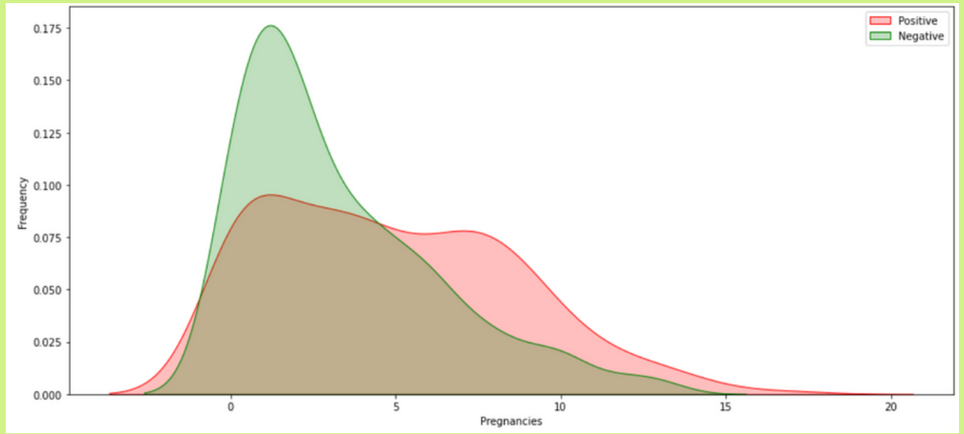


Pada umumnya, semua fitur memiki nilai korelasi yang rendah terhadap variabel Outcome. Namun terdapat beberapa fitur predictor yang memiliki nilai korelasi yang cukup tinggi, yakni :

- Pregnancies dengan Age (Korelasi +) = 0.54 **(STRONG)**
- Glucose dengan Insulin (Korelasi +) = 0.35 **(LOW)**
- BloodPressure dengan BMI (Korelasi +) = 0.28 **(LOW)**
- Glucose dengan Outcome (Korelasi +) = 0.49 **(MEDIUM)**
- SkinThickness dengan BMI (Korelasi +) = 0.55 **(STRONG)**
- BloodPressure dengan Age (Korelasi +) = 0.32 **(LOW)**
- Age dengan Glucose (Korelasi +) = 0.27 **(LOW)**
- BMI dengan Outcome (Korelasi +) = 0.31 **(LOW)**
- SkinThickness dengan Insulin (Korelasi +) = 0.25 **(LOW)**

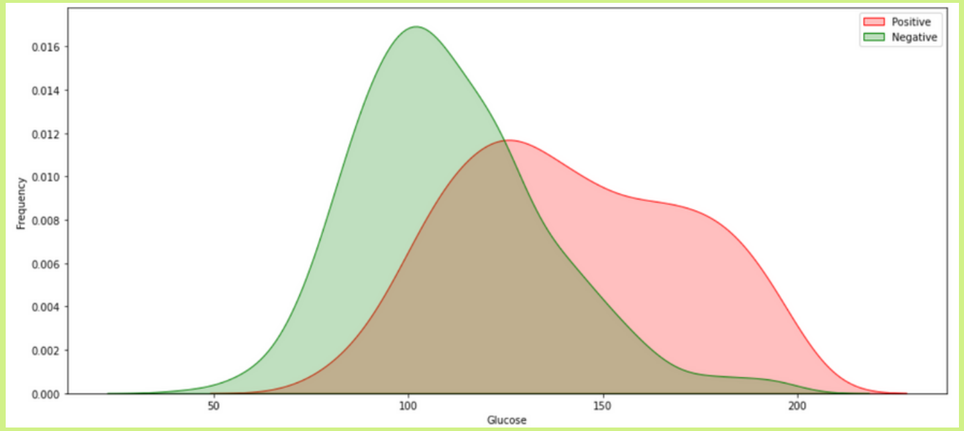
EXPLATORY DATA ANALYSIS (EDA)

Feature VS Outcome



Pregnancies vs Outcome

orang yang positif diabetes cenderung memiliki **riwayat kehamilan (jumlah kehamilan) yang cukup tinggi (> 20 Kali)**. Pasien diabetes paling banyak pada **jumlah kehamilan 2 kali**



Glucose vs Outcome

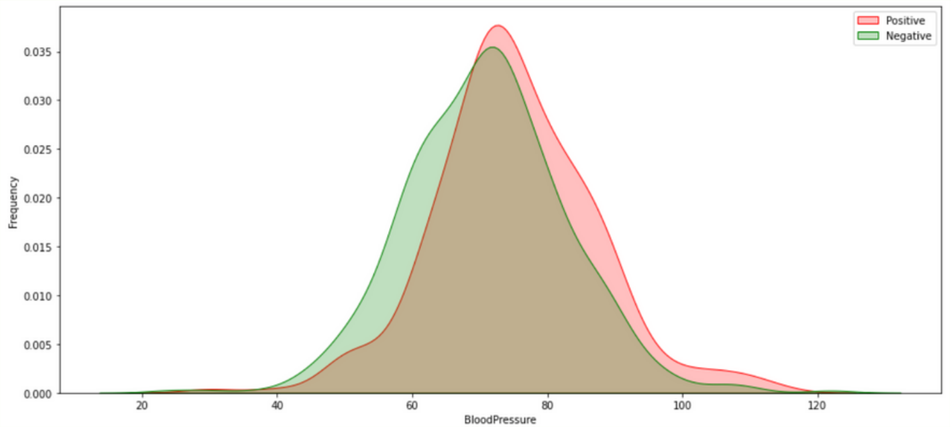
orang yang terkena diabetes memiliki **nilai glucose yang lebih tinggi** daripada orang-orang yang **non-diabetes.** Penderita diabetes paling banyak pada **nilai glucose > 125**

POSITIVE

NEGATIVE

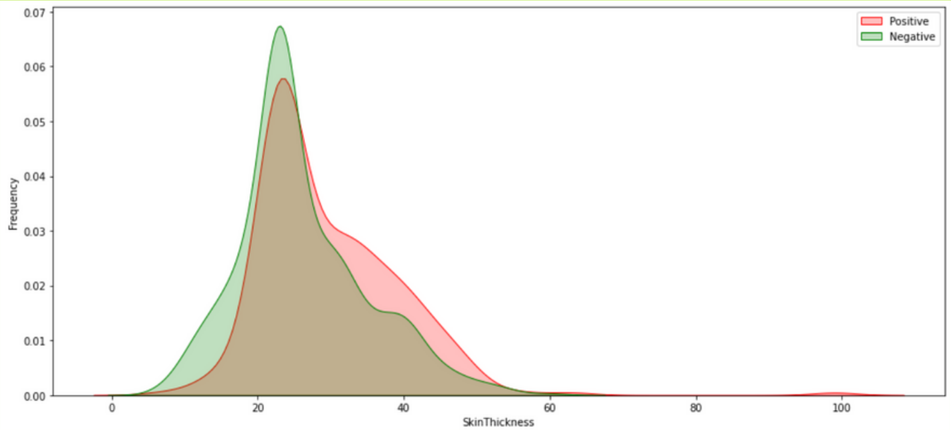
EXPLATORY DATA ANALYSIS (EDA)

Feature VS Outcome



BloodPressure vs Outcome

orang yang terkena diabetes memiliki **nilai bloodPressure yang lebih tinggi** daripada orang-orang yang **non-diabetes.** Terbanyak diderita oleh pasien dengan **bloodPressure >75an.**



SkinThickness vs Outcome

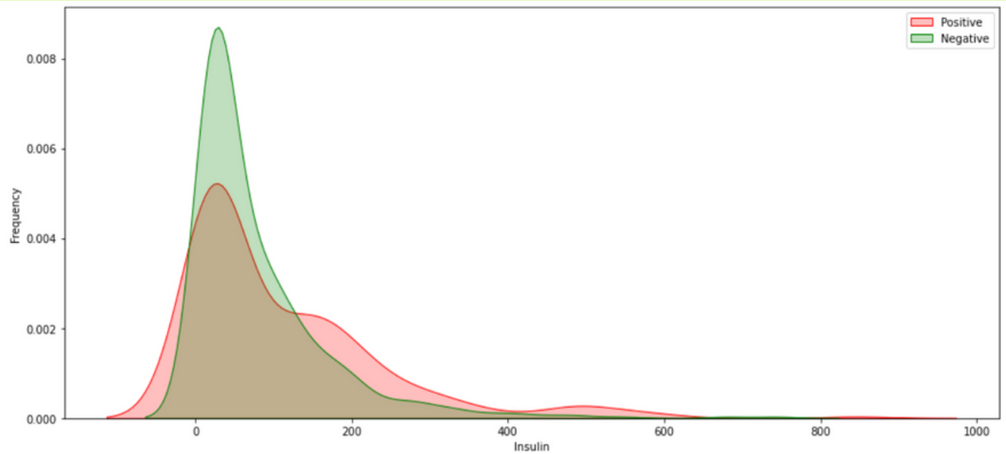
orang yang positif diabetes cenderung memiliki **ketebalan kulit (skinthickness) yang cukup tinggi (hingga nilai 50) .** Range skinThickness diabetes **25-40an** dan paling banyak **dititik 25an.**

POSITIVE

NEGATIVE

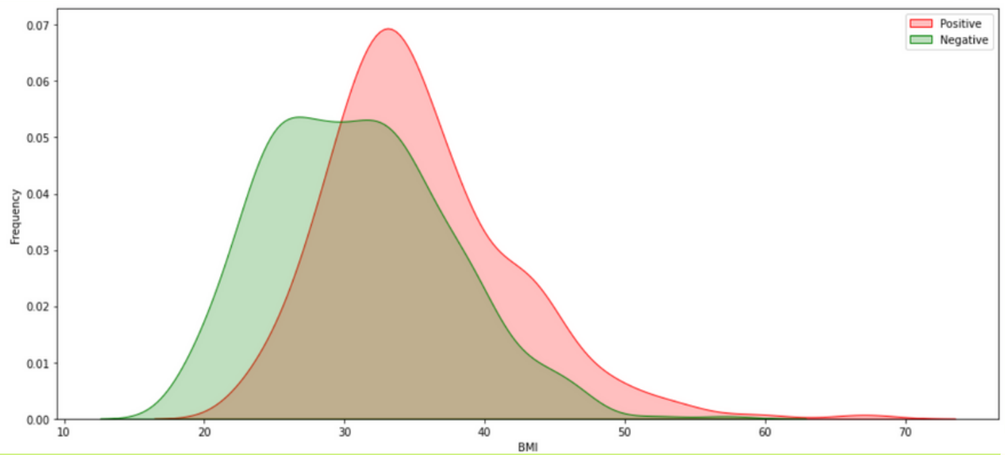
EXPLATORY DATA ANALYSIS (EDA)

Feature VS Outcome



Insulin vs Outcome

orang yang terkena diabetes memiliki insulin berada pada range **0-600an** dan paling banyak diderita oleh orang dengan **range insulin 0-170an**.



BMI vs Outcome

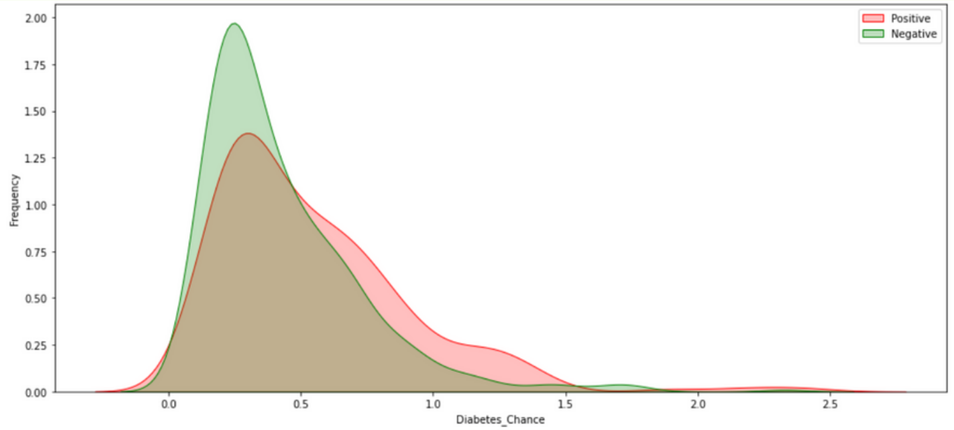
orang yang terkena diabetes memiliki nilai BMI pada **range 20-70 an** dan paling banyak diderita oleh pasien dengan BMI pada range **30-40** artinya sudah masuk kategori **overweight/obese**

POSITIVE

NEGATIVE

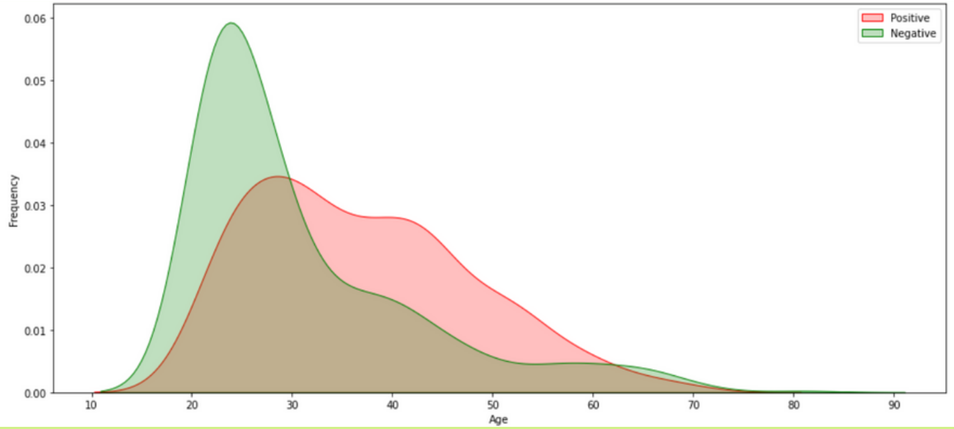
EXPLATORY DATA ANALYSIS (EDA)

Feature VS Outcome



DiabetesChance vs Outcome

orang yang positif diabetes memiliki nilai diabetesChance pada **kisaran 0-2.5**. Paling banyak diderita oleh pasien dengan nilai DiabetesChance pada **range 0-0.5**. Semakin **tinggi** nilai diabetesChance maka semakin **besar peluang** orang tersebut **positif diabetes.**



Age vs Outcome

orang yang terkena diabetes terbanyak berada pada **rentang usia 20-50 tahun.**

POSITIVE

NEGATIVE

FEATURE ENGINEERING

DigitalSkola

Create New Features

```python
def bmi_status(row):
    if row['BMI']<18.5:
        return 'underweight'
    elif row ['BMI'] == 18.5 and row['BMI'] <25:
        return 'healthy'
    elif row['BMI'] == 25 and row['BMI']<30:
        return 'overweight'
    else:
        return 'obese'
```

Menambahkan kolom kategori pada BMI. Kategori dibagi : underweight, ideal/healthy, overweight, obese

```python
def blood_press(row):
    if row['BloodPressure'] <90:
        return 'Low'
    elif row['BloodPressure'] == 90 and row['BloodPressure'] <120:
        return 'Normal'
    elif row['BloodPressure'] == 120 and row['BloodPressure'] <130:
        return 'Elevated'
    elif row['BloodPressure'] == 130 and row['BloodPressure'] <140:
        return 'High Stage 1'
    elif row['BloodPressure'] == 140 and row['BloodPressure'] <180:
        return 'High Stage 2'
    else:
        return 'Hypertensive'
```

Menambahkan kolom kategori dari nilai BloodPressure

FEATURE ENGINEERING

DigitalSkola

Create New Features

```
def glucose_level(row):
    if row['Glucose'] == 70 and row['Glucose'] < 100:
        return 'Normal'
    elif row['Glucose'] == 100 and row['Glucose'] <126:
        return 'Symptomatic'
    elif row['Glucose'] <70:
        return 'Hypoglicemic'
    elif row['Glucose'] == 126 and row['Glucose']<=180:
        return 'Diabetic'
    else:
        return 'Hyperglycemic'
```

Menambahkan kolom kategori dari nilai glucose dibagi ke dalam normal, gejala, hypoglicemic, hyperglicemic, dan diabetic.

```
def test_result(row):
    if row['Outcome']==1:
        return 'Positive'
    else:
        return 'Negative'

def is_obese(row):
    if row['BMI']>=30:
        return 'Yes'
    else:
        return 'No'
```

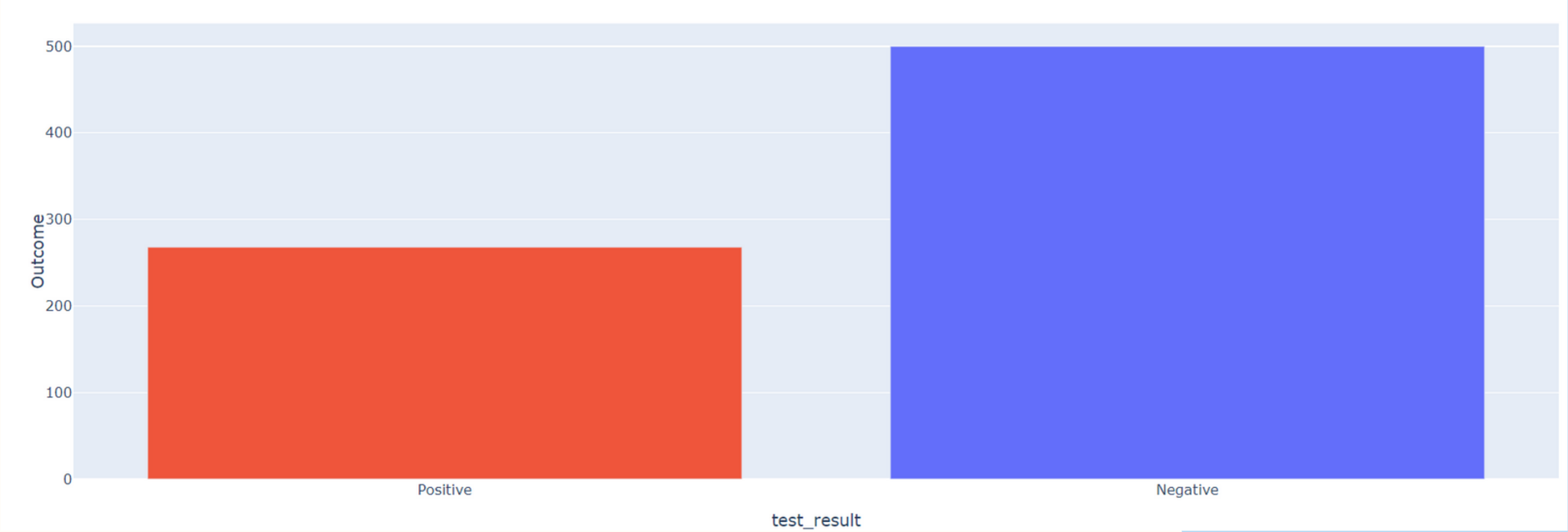
Menambahkan kolom kategori dari kolom Outcome dengan nilai Positive & Negative dan kolom kategori dari BMI Obesitas dengan nilai Yes & No.

DigitalSkola

EDA Test_Result

	test_result	Outcome
0	Negative	500
1	Positive	268

Dari data yang ada, dapat disimpulkan jika **500 orang negatif diabetes** dan **268 orang** lainnya **positif diabetes**.

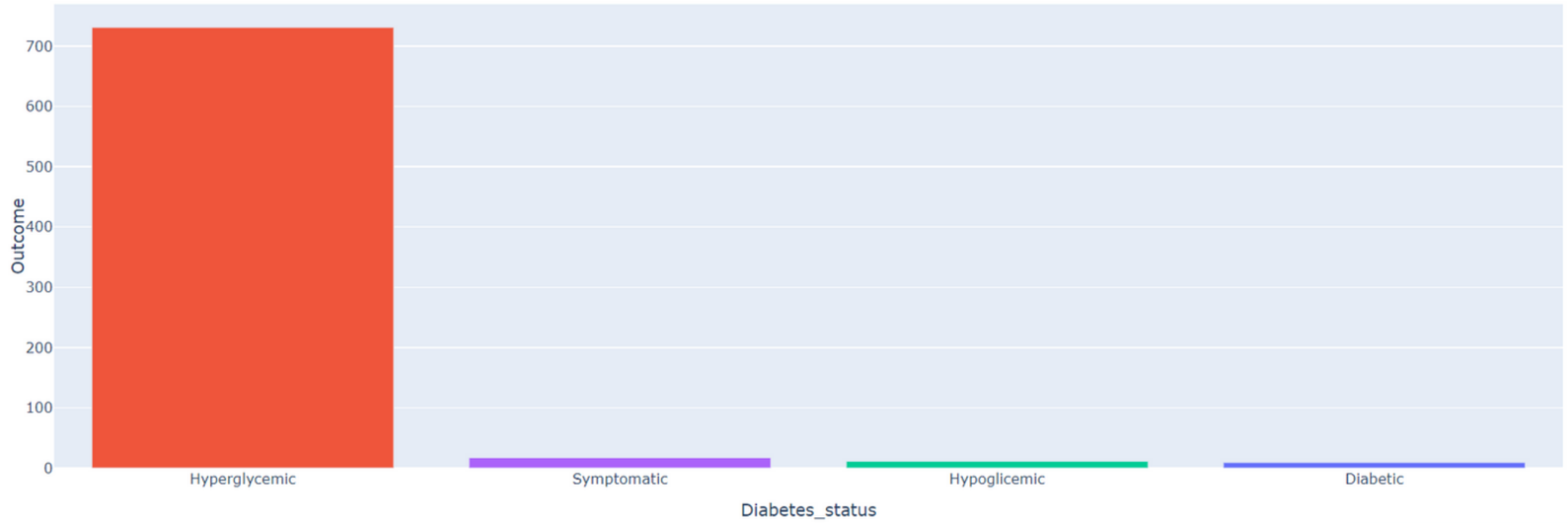


DigitalSkola

EDA Diabetes_Status

	Diabetes_status	Outcome
0	Diabetic	9
1	Hyperglycemic	731
2	Hypoglicemic	11
3	Symptomatic	17

Dari data yang diberikan, jumlah orang yang tergolong **Diabetic (gula darah tinggi)** adalah **9 orang**, orang yang **symptomatic (memiliki tingkat gula darah lebih tinggi dari normal) berjumlah 17 orang**, dan **11 orang** terindikasi **Hypoglicemic (Gula darah di bawah normal)**. **731 Orang** lainnya terindikasi **Hyperglycemic (gula darah ekstrim)**.

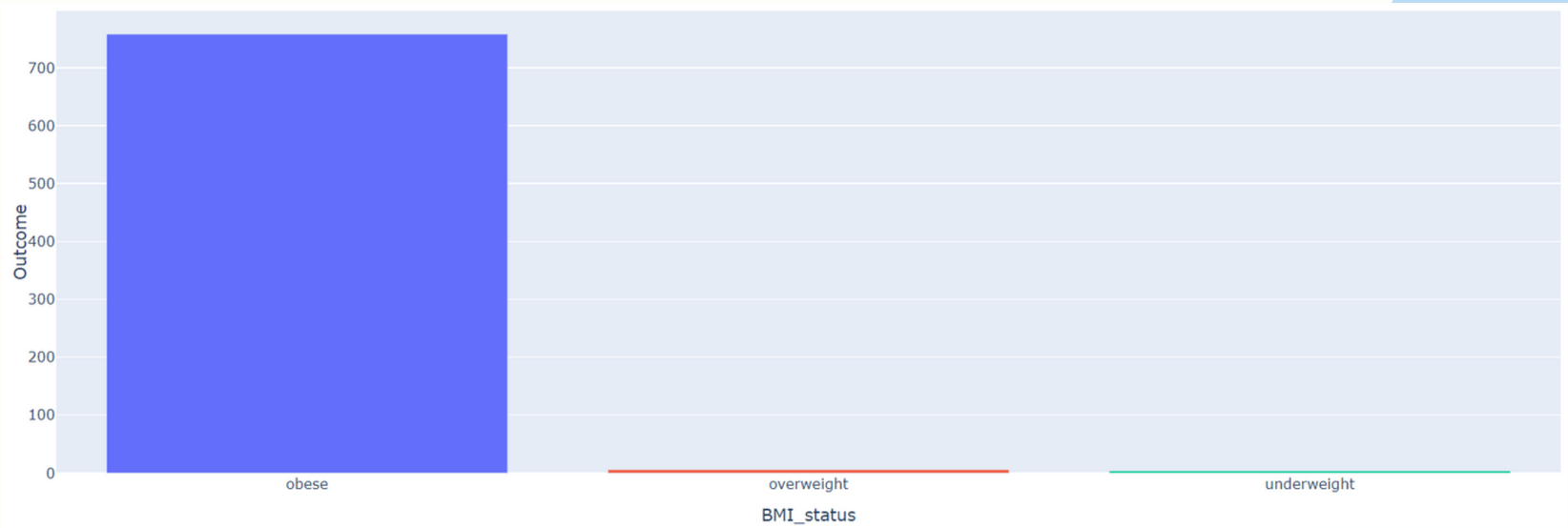


DigitalSkola

EDA BMI_Status

	BMI_status	Outcome
0	obese	758
1	overweight	6
2	underweight	4

Dari data yang diberikan, sangat didominasi oleh orang yang **obesitas yakni 758 orang**. Sangat tidak imbang terhadap status **overweight (6 orang)** dan **underweight (4 orang)**



DigitalSkola

EDA Pressure_Status

	Pressure_status	Outcome
0	Hypertensive	38
1	Low	708
2	Normal	22

Melalui data yang diberikan, mayoritas adalah mereka yang memiliki **tekanan darah rendah sebanyak 708** orang. Ada **38 orang** yang terindikasi **hipertensi** dan **22 orang** yang memiliki **tekanan darah normal**.

FEATURE SCALING

Comparing Feature Scaling Methods

```
#Compare feature scaling methods

scaled_compare = data.copy()

#standard scaler
ss = StandardScaler()
scaled_compare['Glucose_standard_scaler']= ss.fit_transform(scaled_compare[['Glucose']])
scaled_compare['Diabeteschance_standard_scaler']= ss.fit_transform(scaled_compare[['Diabetes_Chance']])

#robust scaler
rs = RobustScaler()
scaled_compare['Glucose_robust_scaler']= rs.fit_transform(scaled_compare[['Glucose']])
scaled_compare['Diabeteschance_robust_scaler']= rs.fit_transform(scaled_compare[['Diabetes_Chance']])

#minmax scaler
mms = MinMaxScaler()
scaled_compare['Glucose_minmax_scaler']= mms.fit_transform(scaled_compare[['Glucose']])
scaled_compare['Diabeteschance_minmax_scaler']= mms.fit_transform(scaled_compare[['Diabetes_Chance']])


scaled_compare.head()
```

is_obese	is_diabetic	Glucose_standard_scaler	Diabeteschance_standard_scaler	Glucose_robust_scaler	Diabeteschance_robust_scaler	Glucose_minmax_scaler	Diabeteschance_minmax_scaler
Yes	Yes	0.866045	0.468492	0.765432	0.665359	0.670968	0.234415
No	No	-1.205066	-0.365061	-0.790123	-0.056209	0.264516	0.116567
No	Yes	2.016662	0.604397	1.629630	0.783007	0.896774	0.253629
No	No	-1.073567	-0.920763	-0.691358	-0.537255	0.290323	0.038002
Yes	Yes	0.504422	5.484909	0.493827	5.007843	0.600000	0.943638

Perbedaan metode scaling dapat terlihat pada tabel diatas. Pada kasus kali ini, kami akan menggunakan "Robust Scaler". **Robust scaler** dipilih karena dianggap menghasilkan modeling yang lebih baik dibanding metode lainnya dan bersifat robust terhadap outlier.

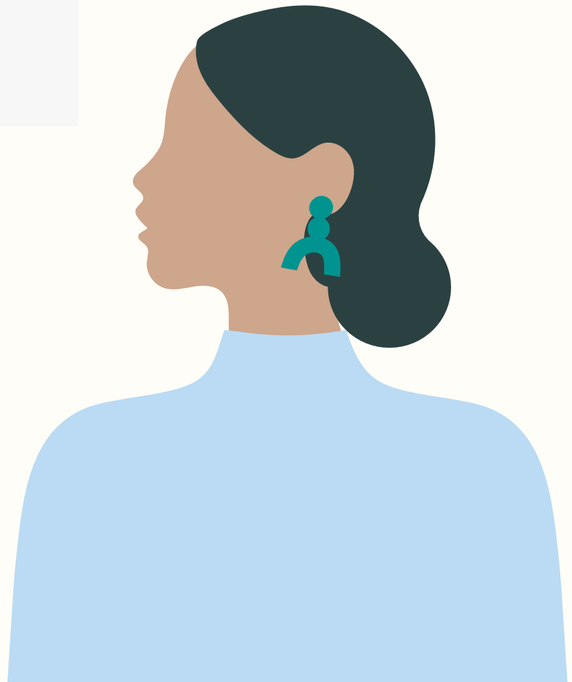
FEATURE SCALING

DigitalSkola

Robust Scaler

```
data_scaled['preg_scaled'] = robust.fit_transform(data_scaled[['Pregnancies']])
data_scaled['chance_scaled'] = robust.fit_transform(data_scaled[['Diabetes_Chance']])
data_scaled['age_scaled'] = robust.fit_transform(data_scaled[['Age']])
data_scaled['skin_scaled'] = robust.fit_transform(data_scaled[['SkinThickness']])
data_scaled['insulin_scaled'] = robust.fit_transform(data_scaled[['Insulin']])
data_scaled['glucose_scaled'] = robust.fit_transform(data_scaled[['Glucose']])
data_scaled['blood_scaled'] = robust.fit_transform(data_scaled[['BloodPressure']])
data_scaled['bmi_scaled'] = robust.fit_transform(data_scaled[['BMI']])
```

Mentransformasi semua tabel numerical agar berada di skala yang sama dengan Robust Scaler.



FEATURE ENCODING

DigitalSkola

Transform Categorical Feature to Numerical with OneHot Encoder

```
#Defining the encoder
enc = OneHotEncoder(handle_unknown='ignore')
```

```
col = sorted(data['is_obese'].unique().tolist()) + sorted(data['is_diabetic'].unique().tolist())
```

```
#Encoding the features
enc_df = pd.DataFrame(enc.fit_transform(data_scaled[['is_obese', 'is_diabetic']]).toarray(), columns=col)
```

```
#Joining the scaled dataframe with encoded dataframe
data_scaled = data_scaled.join(enc_df)
data_scaled.head()
```

```
#creating dummies
index_data = pd.get_dummies(data_scaled[['is_obese', 'is_diabetic']], drop_first=True)
```

```
# joining dummies to the dataframe
final_data=data_scaled.join(index_data)
final_data.head()
```

```
#dropping unused features
final_data.drop(columns=['is_obese','is_diabetic','No','Yes'],axis=1,inplace=True)
```

Transform kolom is_obese dan is_diabetic menjadi numerical kolom.

is_obese_Yes	is_diabetic_Yes
1	1
0	0
0	1
0	0
1	1

DigitalSkola

Oversampling with SMOTE

Balancing akan menggunakan teknik **oversampling menggunakan SMOTE**. Hal ini bertujuan untuk **menyeimbangkan** data positif diabetes dengan negatif diabetes (menambah variasi dari data).

```
#separating independent from dependent variables
X = final_data[['glucose_scaled','insulin_scaled','bmi_scaled','chance_scaled','age_scaled','is_obese_Yes']]
y = final_data['Outcome']
```

```
] X_train.shape
```

```
(576, 6)
```

```
] y_train.shape
```

```
(576,)
```

```
y_train.value_counts()
```

```
0    370
1    206
Name: Outcome, dtype: int64
```

```
print('Before Oversampling: ',Counter(y_train))
#defining smote
SMOTE = SMOTE(random_state=0)

#fit and apply the transform
X_train_smote, y_train_smote = SMOTE.fit_resample(X_train,y_train)

#summarize class distribution
print('After Oversampling: ',Counter(y_train_smote))
```

```
Before Oversampling:  Counter({0: 370, 1: 206})
After Oversampling:  Counter({0: 370, 1: 370})
```

Sekarang data 0 (negatif diabetes) dengan 1 (positif diabetes sudah seimbang.

MODELING & EVALUATION

DigitalSkola

No	Machine Learning Model	Accuracy (F1-Score)	Recall (Outcome =0)	Recall (Outcome =1)	Precision (Outcome =1)	Macro Average (F1-Score)
1	Decision Tree Classifier (No Resampling)	73%	79%	57%	55%	68%
2	Decision Tree Classifier (With Resampling)	73%	75%	68%	54%	70%
3	KNN (No Resampling)	77%	84%	62%	63%	73%
4	KNN (With Resampling)	73%	68%	83%	53%	71%
5	SVC Linear (No Resampling)	83%	92%	64%	77%	79%
6	SVC Linear (With Resampling)	75%	78%	68%	57%	72%
7	SVC RBF (No Resampling)	80%	90%	57%	71%	75%
8	SVC RBF (With Resampling)	81%	79%	85%	63%	79%
9	Random Forest Classifier (No Resampling)	82%	89%	66%	72%	78%
10	Random Forest Classifier (With Resampling)	77%	79%	72%	61%	74%
11	GaussianNB (No Resampling)	78%	87%	57%	66%	73%
12	GaussianNB (With Resampling)	77%	81%	68%	62%	74%
13	Logistic Regression (No Resampling)	82%	90%	64%	73%	78%
14	Logistic Regression (With Resampling)	76%	79%	70%	59%	73%

THE CODE USED TO MODEL

SVC RBF

In [129]:

```
# SVCR without resampling
svc3 = SVC(kernel='rbf',random_state=0)
svc3.fit(X_train,y_train)
```

SVC

```
SVC(random_stat
e=0)
```

```
# SVCR with resampling
svc4 = SVC(kernel='rbf',random_state=0)
svc4.fit(X_train_smote,y_train_smote)
```

SVC

```
SVC(random_stat
e=0)
```

MODEL EVALUATION CODE

DigitalSkola

```
53]: y_pred_svc4 = svc4.predict(X_test)
     print(classification_report(y_test, y_pred_svc4))
```

```
              precision    recall  f1-score   support

           0       0.92      0.79      0.85       107
           1       0.63      0.85      0.73        47

    accuracy                           0.81       154
   macro avg       0.78      0.82      0.79       154
weighted avg       0.84      0.81      0.81       154
```

HYPERPARAMETER TUNING

GridSearch CV

DigitalSkola

Sebelum Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.92	0.79	0.85	107
1	0.63	0.85	0.73	47
accuracy			0.81	154
macro avg	0.78	0.82	0.79	154
weighted avg	0.84	0.81	0.81	154

Setelah Hyperparameter Tuning

```
grid_search.best_params_
```

```
{'C': 1000000, 'gamma': 0.001}
```

	precision	recall	f1-score	support
0	0.92	0.78	0.84	107
1	0.62	0.85	0.72	47
accuracy			0.80	154
macro avg	0.77	0.81	0.78	154
weighted avg	0.83	0.80	0.81	154

HYPERPARAMETER TUNING

RandomizedSearch CV

Sebelum Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.92	0.79	0.85	107
1	0.63	0.85	0.73	47
accuracy			0.81	154
macro avg	0.78	0.82	0.79	154
weighted avg	0.84	0.81	0.81	154

Setelah Hyperparameter Tuning

```
random_search.best_params_
```

```
{'kernel': 'rbf', 'gamma': 0.001, 'C': 1000000}
```

	precision	recall	f1-score	support
0	0.92	0.78	0.84	107
1	0.62	0.85	0.72	47
accuracy			0.80	154
macro avg	0.77	0.81	0.78	154
weighted avg	0.83	0.80	0.81	154

HYPERPARAMETER TUNING

GridSearch CV VS RandomizedSearch CV

GridSearchCV
Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.92	0.78	0.84	107
1	0.62	0.85	0.72	47
accuracy			0.80	154
macro avg	0.77	0.81	0.78	154
weighted avg	0.83	0.80	0.81	154

RandomizedSearchCV
Hyperparameter Tuning

	precision	recall	f1-score	support
0	0.92	0.78	0.84	107
1	0.62	0.85	0.72	47
accuracy			0.80	154
macro avg	0.77	0.81	0.78	154
weighted avg	0.83	0.80	0.81	154

HYPERPARAMETER TUNING

GridSearch CV VS RandomizedSearch CV

Bahwa performa model lebih baik tanpa di hyperparameter tuning, karena setelah di hyperparameter tuning baik dengan GridSearch maupun RandomizedSearch hasilnya sedikit menurun. Hasil terbaik dengan nilai berikut :

	precision	recall	f1-score	support
0	0.92	0.78	0.84	107
1	0.62	0.85	0.72	47
accuracy			0.80	154
macro avg	0.77	0.81	0.78	154
weighted avg	0.83	0.80	0.81	154

Hasil tanpa hyperparameter tuning

Conclusion

Setelah menggunakan dataset pasien ini, kami dapat membangun model machine learning (support vector machine classifier with kernel rbf – yang terbaik) untuk secara akurat memprediksi apakah pasien dalam kumpulan data menderita diabetes atau tidak, dan dengan itu kami dapat menarik beberapa wawasan dari data melalui analisis data dan visualisasi.

DigitalSkola

Suggestion

Dengan dataset yang ada kami mungkin dapat mendeteksi apakah seseorang menderita diabetes atau tidak, tetapi kami tidak dapat memprediksi jenis diabetes menggunakan dataset tersebut. Untuk kedepannya kami berharap dapat dilakukan penelitian lebih lanjut guna memprediksi jenis diabetes dan mengeksplorasi proporsi masing-masing indikator, yang dapat meningkatkan akurasi prediksi diabetes.

DigitalSkola

thank you