



FINAL PROJECT





**ARIZAL NUR
ISLACHUDDIN**

0110219034

RISIKO KREDIT

Risiko kredit merupakan area risiko kritis dalam bisnis perbankan. Jika tidak dikelola secara efektif, itu menyebabkan kredit macet atau aset buruk, mengurangi margin keuntungan bank, mengikis modal dan dalam kasus ekstrim, dapat menyebabkan kegagalan bank.

Bank Talenta Rakyat adalah bank yang menangani kredit rakyat dan ingin menggunakan pembelajaran mesin untuk mengoptimalkan proses screening profil nasabah yang mengajukan kredit. Ketika bank meminjamkan uang kepada pelanggan, itu mengambil semacam risiko. Jadi, sebelum menyetujui pinjaman, bank akan memeriksa apakah peminjam akan memiliki cukup uang dimasa depan untuk membayar kembali pinjaman mereka. Berdasarkan pendapatan dan riwayat kredit pelanggan saat ini, Bank Talenta Rakyat melakukan semacam analisis yang membantu mereka memutuskan apakah peminjam akan menjadi pelanggan yang baik untuk bank itu atau tidak.



Langkah-langkah Teknis



- Import library
- Read Dataset
- Data Preparation
- Exploratory Data Analysis
- Model Development
- Model Deployment
- Sumarry

01

Import Library

Library Python adalah kumpulan kode pada Python yang dapat digunakan kembali dalam beberapa program atau proyek.

- Pandas

Digunakan untuk melakukan pre-processing dan analisis data.

- Numpy

Digunakan dalam membantu proses komputasi numerik

- Matplotlib

Digunakan untuk menyajikan data ke dalam bentuk visual.

- Seaborn

Digunakan untuk membuat grafik visualisasi data.

- Scikit-learn

Digunakan untuk membuat model machine learning.

- Imbalearn

Digunakan untuk mengatasi data yang tidak seimbang

- Statsmodel

Digunakan untuk mengaplikasikan berbagai model statistika ke dalam data

Read Dataset

- Melakukan import source data kedalam google colabs melalui google drive.

```
from google.colab import drive  
drive.mount('/content/drive')
```

- Membaca file kedalam bentuk dataframe.

1. membaca dataset male :

```
df_1=pd.read_csv
```

2. membaca dataset female :

```
df_2=pd.read_csv
```

3. membaca dataset credit history :

```
df_credit=pd.read_csv
```



03

Data Preparation

a. Data Transformation

- Menggabungkan dataframe male dan dataframe female menjadi dataframe applicant menggunakan perintah concat
- menghapus data duplikat pada data frame credit history dengan mengambil value maksimal pada status overdue.
- Menggabungkan dataframe applicant dengan data credit history dengan menggunakan metode inner join. menjadi data applicant history dimana ini adalah data yang akan menjadi data utama dalam analisis dan pembuatan model

b. Data Preprocesing

- Melihat dan merubah tipe data yang salah pada kolom data applicant history
(Object => Float => Integer) : "Pendapatan"
(Object=>Boolean): "KepemilikanMobil", "KepemilikanProperti"
(Int=>Boolean): "FlagMobile", "FlagWorkPhone", "FlagPhone", "Email"
- merubah nama value pada beberapa kolom kategori data applicant history
(Tipe Pendidikan) = "PG" : "Post Graduate", "G": "Graduate", "UG": "Undergraduate"
(Status Keluarga) = "M" : "Menikah", "NM" : "Belum Menikah", "D" : "Cerai"
(Status Keluarga) = "RA" : "Sewa Apartment", "MH" : "Rumah Pribadi", "MA" :
"Apartment Pribadi", "PH" : "Rumah Orang Tua", "OA" : "Apartemen Kantor"





- cek missing values

Pendapatan	45
Pekerjaan	11357

Pada tahap ini dilakukan handling missing values pada kolom yang terdapat missing value dengan menghapus missing value tersebut

- drop values dari kolom overdue yang tidak digunakan untuk membuat model ('tidak memiliki pinjaman')
- Labeling good score (0) & bad score (1) pada variabel target yaitu kolom overdue status dengan menggunakan looping

Sebelum labeling

0	19543
1	2614
2	212
5	124
3	42
4	37

Sesudah labeling

0	19543
1	3029

04

Exploratory Data Analysis

Menampilkan Baris Paling Atas
"df_applicant_history.head()"

Menampilkan Jumlah Baris & Kolom
"df_applicant_history.shape"

Menampilkan Nama Kolom
"df_applicant_history.columns"

Menampilkan Jumlah Nilai Unik pada Data
"df_applicant_history.nunique()"

Menampilkan Jumlah Nilai Unik pada Variabel Target
"df_applicant_history['Overdue_status'].value_counts()"

Menampilkan Informasi Mengenai Data
"df_applicant_history.info()"

Melihat Kesimpulan Data Secara Statistik
"df_applicant_history.iloc[:,1:].describe()"

- menampilkan lima baris teratas pada dataset applicant history

	Id_customer	JK	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	TipePendapatan	TingkatPendidikan	StatusKeluarga	TipeRumah	FlagMobile	FlagWorkPhone	FlagPhone	Email	Pekerjaan	JmlAnggotaKeluarga	Age	Experience
0	5008806	Laki-laki	True	True	0	112500	Bekerja	Graduate	Menikah	Rumah Pribadi	True	False	False	False	Security staff	2	59	3
1	5008815	Laki-laki	True	True	0	270000	Bekerja	Post Graduate	Menikah	Rumah Pribadi	True	True	True	True	Accountants	2	46	2
2	5112956	Laki-laki	True	True	0	270000	Bekerja	Post Graduate	Menikah	Rumah Pribadi	True	True	True	True	Accountants	2	46	2
3	5008820	Laki-laki	True	True	0	135000	Asosiasi komersial	Graduate	Menikah	Rumah Pribadi	True	False	False	False	Laborers	2	49	3
4	5008821	Laki-laki	True	True	0	135000	Asosiasi komersial	Graduate	Menikah	Rumah Pribadi	True	False	False	False	Laborers	2	49	3

- menampilkan nama nama kolom yang terdapat pada dataset applicant history

```
Index(['Id_customer', 'JK', 'KepemilikanMobil', 'KepemilikanProperti',
       'JmlAnak', 'Pendapatan', 'TipePendapatan', 'TingkatPendidikan',
       'StatusKeluarga', 'TipeRumah', 'FlagMobile', 'FlagWorkPhone',
       'FlagPhone', 'Email', 'Pekerjaan', 'JmlAnggotaKeluarga', 'Age',
       'Experience', 'Overdue_status'],
      dtype='object')
```

- Dari dataset applicant history terdapat 22572 baris dan memiliki 18 kolom independen dan 1 kolom dependen yaitu overdue status

(22572, 19)

- Melihat jumlah nilai unik pada tiap kolom dataset applicant history

<code>Id_customer</code>	22572
<code>JK</code>	2
<code>KepemilikanMobil</code>	2
<code>KepemilikanProperti</code>	2
<code>JmlAnak</code>	9
<code>Pendapatan</code>	182
<code>TipePendapatan</code>	5
<code>TingkatPendidikan</code>	3
<code>StatusKeluarga</code>	3
<code>TipeRumah</code>	5
<code>FlagMobile</code>	1
<code>FlagWorkPhone</code>	2
<code>FlagPhone</code>	2
<code>Email</code>	2
<code>Pekerjaan</code>	18
<code>JmlAnggotaKeluarga</code>	10
<code>Age</code>	47
<code>Experience</code>	43
<code>Overdue_status</code>	2
<code>dtype:</code>	<code>int64</code>

- Melihat jumlah nilai unik pada variabel target yaitu overdue status , dimana kelas 0 (good score) berjumlah lebih banyak yaitu 19543 orang dibandingkan kelas 0 (bad score) dengan jumlah 3029 orang

<code>0</code>	19543
<code>1</code>	3029

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22572 entries, 0 to 22571
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Id_customer      22572 non-null   int64  
 1   JK              22572 non-null   object  
 2   KepemilikanMobil 22572 non-null   bool   
 3   KepemilikanProperti 22572 non-null   bool   
 4   JmlAnak         22572 non-null   int64  
 5   Pendapatan       22572 non-null   int64  
 6   TipePendapatan  22572 non-null   object  
 7   TingkatPendidikan 22572 non-null   object  
 8   StatusKeluarga  22572 non-null   object  
 9   TipeRumah        22572 non-null   object  
 10  FlagMobile       22572 non-null   bool   
 11  FlagWorkPhone   22572 non-null   bool   
 12  FlagPhone        22572 non-null   bool   
 13  Email            22572 non-null   bool   
 14  Pekerjaan        22572 non-null   object  
 15  JmlAnggotaKeluarga 22572 non-null   int64  
 16  Age              22572 non-null   int64  
 17  Experience       22572 non-null   int64  
 18  Overdue_status  22572 non-null   int64  
dtypes: bool(6), int64(7), object(6)
memory usage: 2.4+ MB
```

- melihat informasi mengenai tipe data pada dataset applicant history , dimana terdapat 3 jenis tipe data yaitu boolean (6) ,integer(7) dan object (6) dan Tidak ada kolom variabel yang memiliki nilai null/hilang

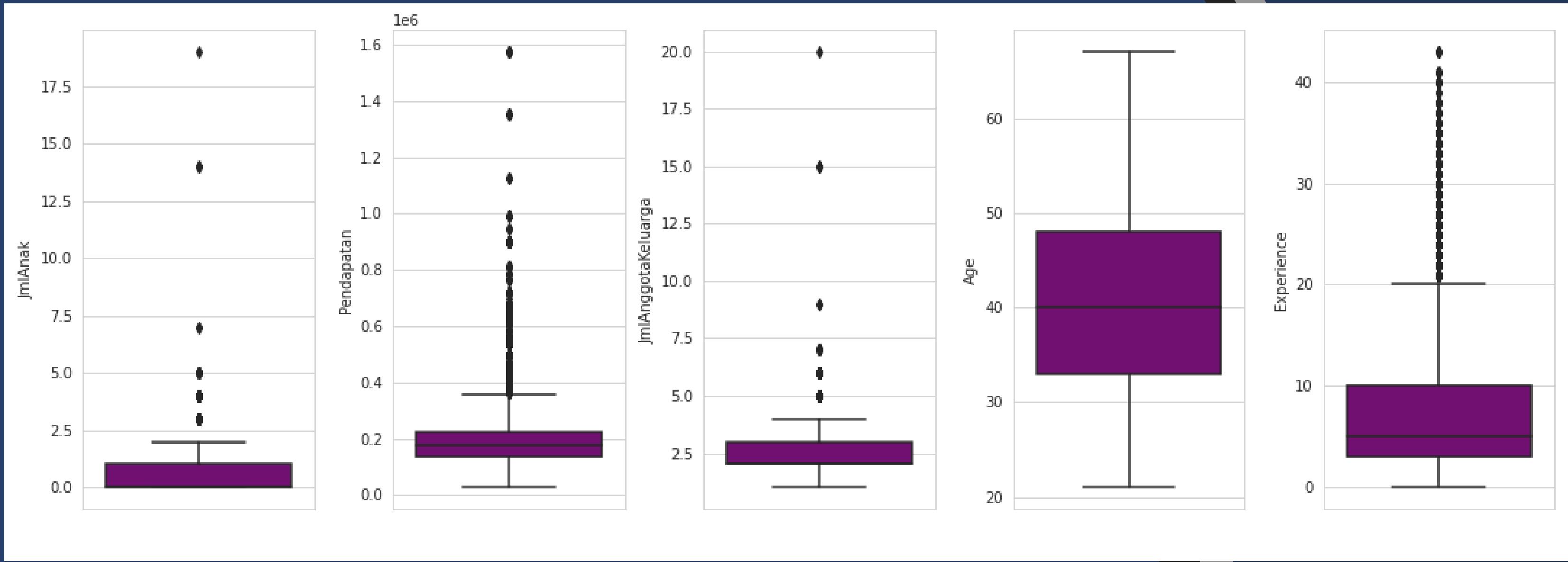
MELIHAT KESIMPULAN DATA SECARA STATISTIK

	JmlAnak	Pendapatan	JmlAnggotaKeluarga	Age	Experience	Overdue_status
count	22572.000000	2.257200e+04	22572.000000	22572.000000	22572.000000	22572.000000
mean	0.511474	1.942669e+05	2.291334	40.604288	7.242158	0.134193
std	0.788750	1.044084e+05	0.948783	9.593509	6.449541	0.340867
min	0.000000	2.700000e+04	1.000000	21.000000	0.000000	0.000000
25%	0.000000	1.350000e+05	2.000000	33.000000	3.000000	0.000000
50%	0.000000	1.800000e+05	2.000000	40.000000	5.000000	0.000000
75%	1.000000	2.250000e+05	3.000000	48.000000	10.000000	0.000000
max	19.000000	1.575000e+06	20.000000	67.000000	43.000000	1.000000

- dapat dilihat ada perbedaan besar pada kuatil atas 75% dengan nilai maksimal pada kolom "jmlAnak", "jmlAnggotaKeluarga", yang memungkinkan bahwa ada nilai outlier ekstrim pada kumpulan data ini

Visualize outlier

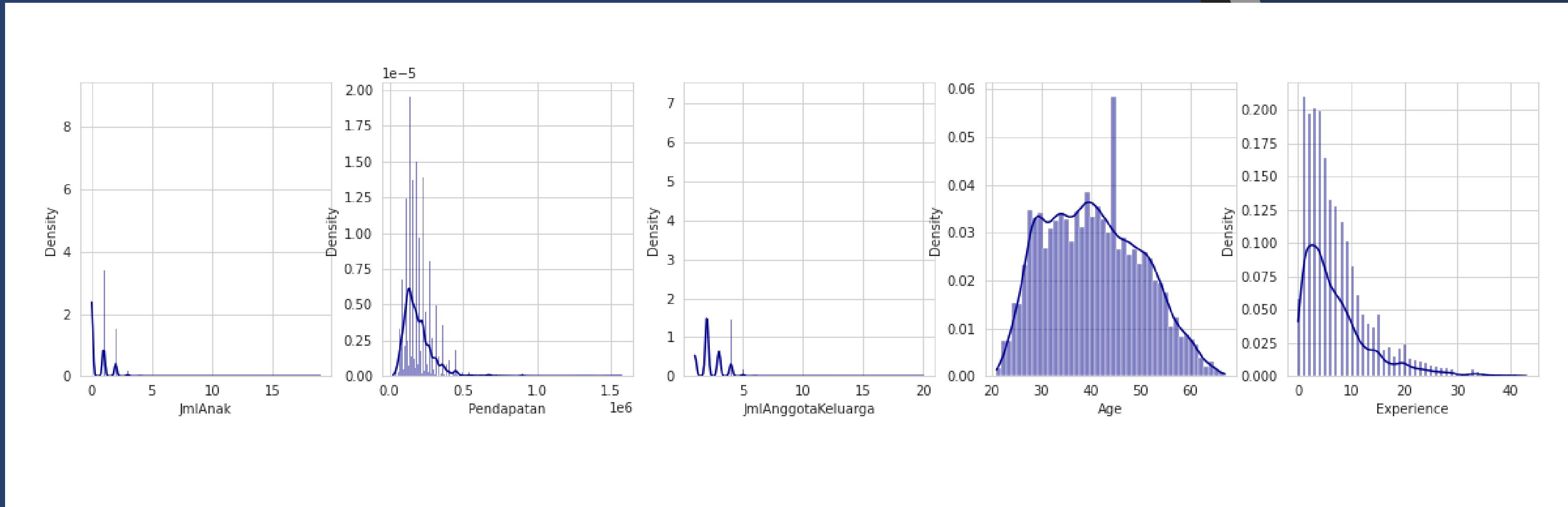
- bentuk visualisasi outlier dimana kotak persegi merupakan sebuah median dan garis tengah yang memiliki ujung diatas dan di bawah disebut "whisker" batas minimum dan maksimum



- dari tampilan outlier diatas pada kolom jumlah anak dan anggota keluarga ada 2 nilai yang jauh dari sebaran data pada umumnya , tetapi tidak semua outlier itu dianggap sebagai sesuatu yang buruk karena pada suatu kondisi itu bukanlah sebuah human error melainkan sebuah nilai yang sesuai dengan kenyataan pada saat menginput data dan merupakan sebuah variasi dari kumpulan data

Visualize Density

- Sekarang untuk memeriksa linearitas variabel, cara yang baik adalah memplot grafik distribusi dan mencari kemiringan fitur. dimana Kernel density estimate (kde) yang t berguna untuk memplot bentuk distribusi.



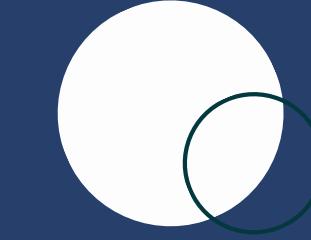
- semua variabel independen miring ke kanan / miring positif, yang berarti nilai rata-rata lebih besar dari median, dan median lebih besar dari modus

KORELASI VARIABEL

	column_name	Overdue_status
0	TipePendapatan_Pensioner	0.060976
1	KepemilikanProperti	0.035332
2	Pendapatan	0.029784
3	Age	0.021222
4	Email	0.021085
5	Jenis_Kelamin	0.020267
6	Pekerjaan_Private service staff	0.019523
7	TipePendapatan_Bekerja	0.018586
8	Pekerjaan_Low-skill Laborers	0.018126
9	Pekerjaan_Sales staff	0.015149

- menentukan variabel manakah yang paling berkorelasi terhadap variabel target, dimana ini merupakan top 10 variabel memiliki nilai korelasi paling tinggi .
- Nilai korelasi yang mendekati -1 atau +1 artinya menyatakan hubungan yang makin kuat. Nilai di atas nol akan menunjukkan korelasi positif, sedangkan nilai di bawah nol berarti menunjukkan korelasi negatif.
- Nilai positif menunjukkan arah hubungan searah. Artinya jika X naik, maka Y naik dan begitu juga sebaliknya.

catatan : dalam hal ini saya membuat korelasinya menjadi positif semua agar bisa disortir dan ditentukan manakah variabel dengan korelasi tertinggi



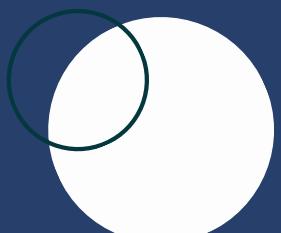
05

MODEL DEVELOPMENT

supaya komputer bisa memahami dataset maka perlu dilakukan one hot encoding atau metode merubah value data menjadi bentuk biner. ini merupakan bentuk dataset setelah dilakukan one hot encoding dimana inilah data yang akan dipakai untuk pembuatan model machine learning

	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	FlagWorkPhone	FlagPhone	Email	JmlAnggotaKeluarga	Age	Experience	Overdue_status	Jenis_Kel
0	True	True	0	112500	False	False	False	2	59	3	0	
1	True	True	0	270000	True	True	True	2	46	2	0	
2	True	True	0	270000	True	True	True	2	46	2	0	
3	True	True	0	135000	False	False	False	2	49	3	0	
4	True	True	0	135000	False	False	False	2	49	3	0	
...	
22567	False	True	0	180000	False	False	False	2	54	10	1	
22568	False	True	0	180000	False	False	False	2	54	10	1	
22569	False	True	0	157500	False	True	True	2	34	4	1	
22570	False	True	0	157500	False	True	True	2	34	4	1	
22571	False	True	0	283500	False	False	False	2	49	2	1	

22572 rows × 46 columns



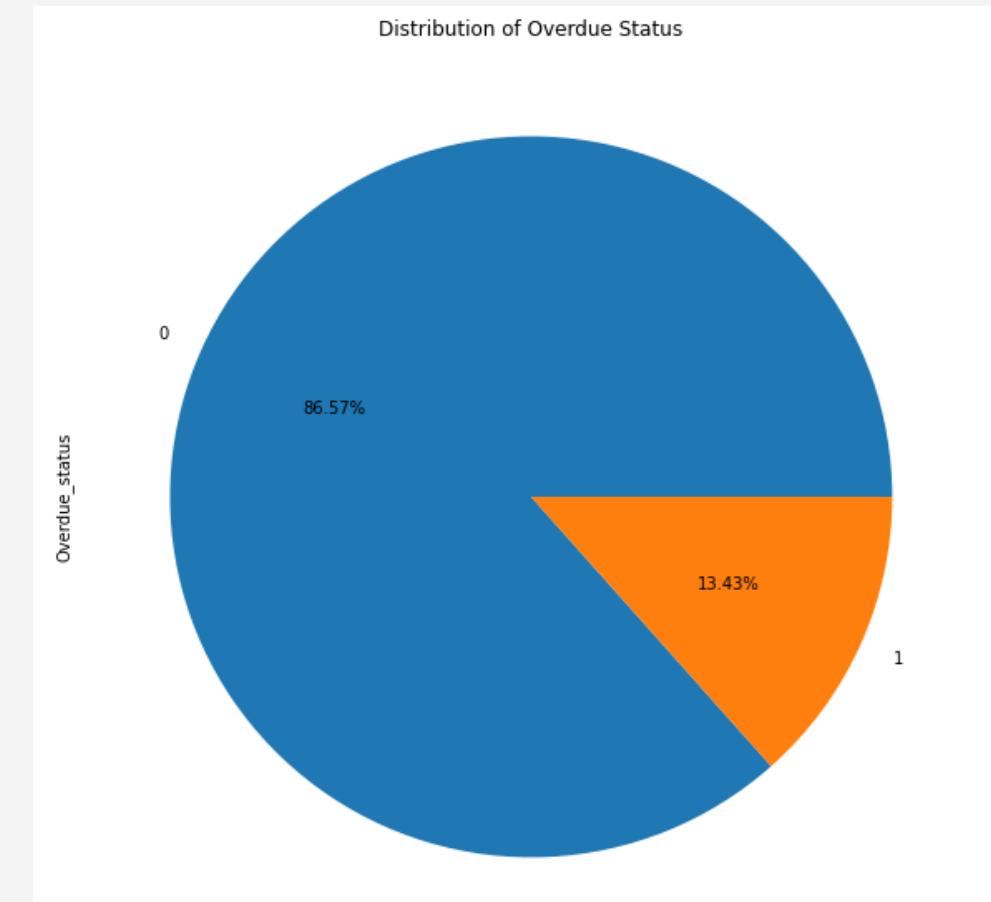
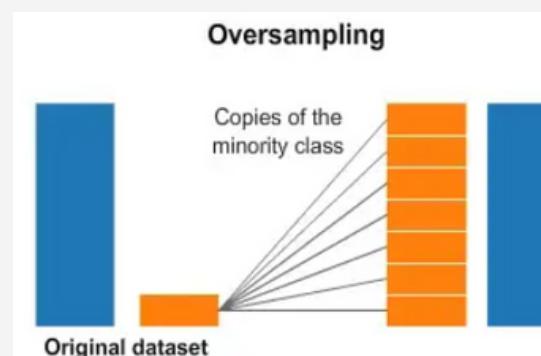


IMBALANCE DATA

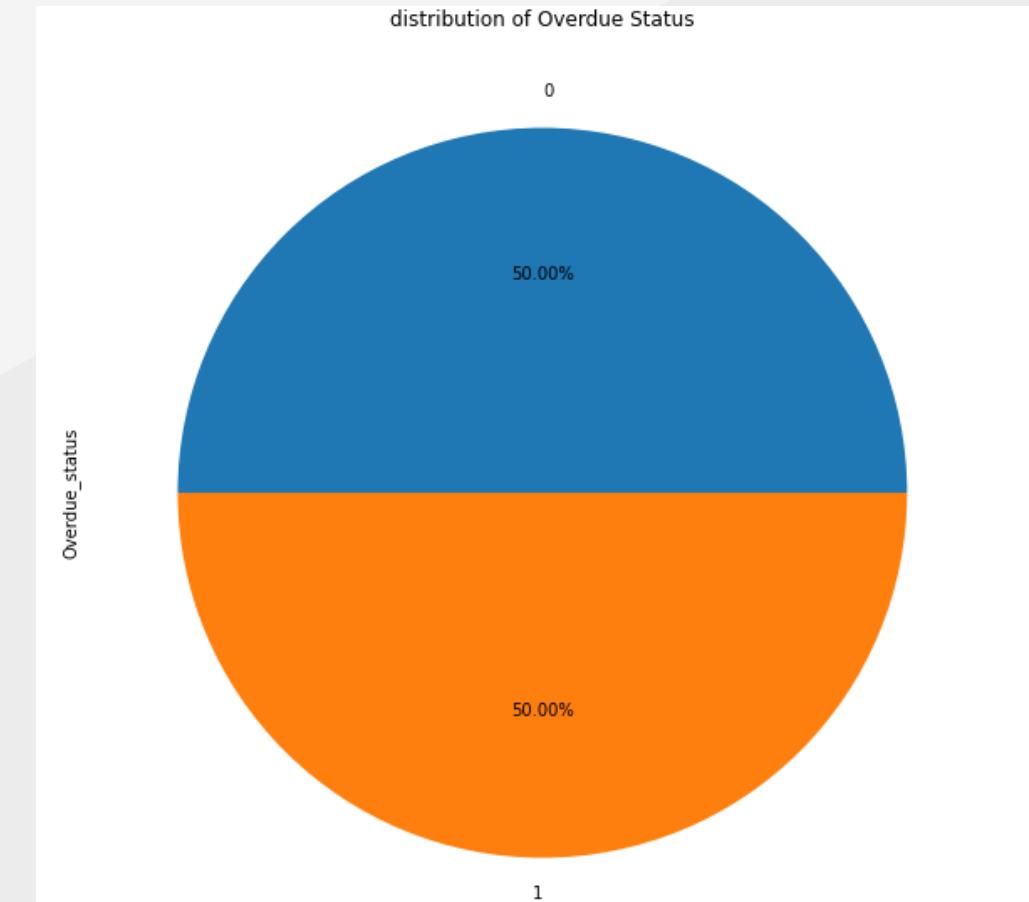
langkah pertama dalam pembuatan model adalah mendefinisikan nilai X , Y. dimana X adalah variabel independen dan Y adalah variabel dependen atau variabel target . kemudian membagi data menjadi data train (untuk pelatihan untuk model) dan data test (untuk melakukan prediksi), perlu diketahui bahwa dataset applicant history memiliki ketimpangan jumlah kelas yang dapat dilihat dari jumlah tiap kelas pada target variabel , ini berbahaya untuk model karena jika dibiarkan maka model bisa salah memprediksi data dengan menganggap rare class sebagai abundant class. maka dari itu perlu dilakukan resampling data pada data train dan menyimpannya pada X dan Y yang baru.

OVERSAMPLING

resampling adalah teknik memanipulasi data untuk menyeimbangkan proporsi kelasnya, dalam hal ini metode yang dipakai adalah oversampling. secara umum metode ini adalah melakukan generate kelas minoritas sebanyak kelas mayoritas sehingga memiliki proporsi yang seimbang .



Sebelum resampling



Setelah resampling data

MODEL SELECTION

REGRESI LOGISTIK

Regresi logistik adalah algoritma supervised machine learning dimana algoritma ini digunakan untuk melakukan klasifikasi berdasarkan hubungan antara variabel dependen dengan variabel target . Di mana, hasilnya harus menjadi nilai kategoris atau diskrit. Itu bisa berupa Ya atau Tidak, 0 atau 1, benar atau Salah

Feature Selection

dari data yang telah dilakukan resampling , selanjutnya adalah memilih fitur terbaik yang akan digunakan untuk pelatihan model dengan mengurangi jumlah fitur yang terlibat dalam menentukan suatu nilai kelas target. Fitur yang diabaikan biasanya berupa fitur yang tidak relevan dan data berlebih.

Model Implementation

mengimplementasikan data kedalam model machine learning , dengan melatih model dan melakukan prediksi

Model Evaluation

melakukan evaluasi terhadap model dan hasil prediksinya dengan melakukan uji statistik

Uji Nilai Statistik Terhadap Fitur

Metode yang digunakan untuk seleksi fitur dimana pada dasarnya merupakan proses rekuksif yang meranking fitur berdasarkan tingkat pentingnya terhadap proses prediksi. dimana dari total 45 fitur hanya 22 fitur yang benar benar penting

STATSMODEL

melakukan uji statistik terhadap 22 fitur terbaik dengan melihat p-value (nilai probabilitasnya) terhadap target variabel. Apa sih maksud dari p-value sebesar 0.05 atau 5% tuh? Sederhananya, p-value sebesar 0.05 itu maksudnya adalah semisal kita melakukan 100 percobaan, maka hipotesis kita akan terjadi sebanyak 95 kali. Artinya, dari 100 percobaan, hanya 5 kali hipotesis kita gagal (gak terjadi). singkatnya Jika p-value kurang dari 0.05, maka hipotesis null (H_0) ditolak, dengan kata lain maka hipotesis kita (H_a) Gagal ditolak (diterima). dapat dilihat bahwa pada 22 fitur p-valuenya kurang dari 5%, dengan kata lain 22 fitur inilah yang akan digunakan untuk melatih model regresi logistik dalam memprediksi variabel target

MODEL IMPLEMENTATION

Mendefinisikan X dan Y yang baru yang mana telah dilakukan resampling dan feature selection kedalam bentuk train test split data

```
"train_test_split(X_lr, y_lr, test_size=0.2, random_state=0)"
```

Melakukan scaling data atau normalisasi data

```
"sc=StandardScaler()"
```

Membuat classifiernya

```
"logreg = LogisticRegression()"
```

Latih model menggunakan data pelatihan

```
"logreg.fit(X_train,y_train)"
```

Prediksi model menggunakan data test

```
"y_pred = logreg.predict(X_test)"
```

```
# membagi data yang telah disiapkan untuk model kedalam data train test split
X_train, X_test, y_train, y_test = train_test_split(X_lr, y_lr, test_size=0.2, random_state=0)

# Scaling data
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
X_train=pd.DataFrame(X_train,columns=X_lr.columns)
X_test=pd.DataFrame(X_test,columns=X_lr.columns)

# buat Classifiernya
# logreg = LogisticRegression(solver='lbfgs', max_iter=1000)
logreg = LogisticRegression()

#latih model menggunakan data pelatihan
logreg.fit(X_train,y_train)

# prediksi model menggunakan data test
y_pred = logreg.predict(X_test)
```

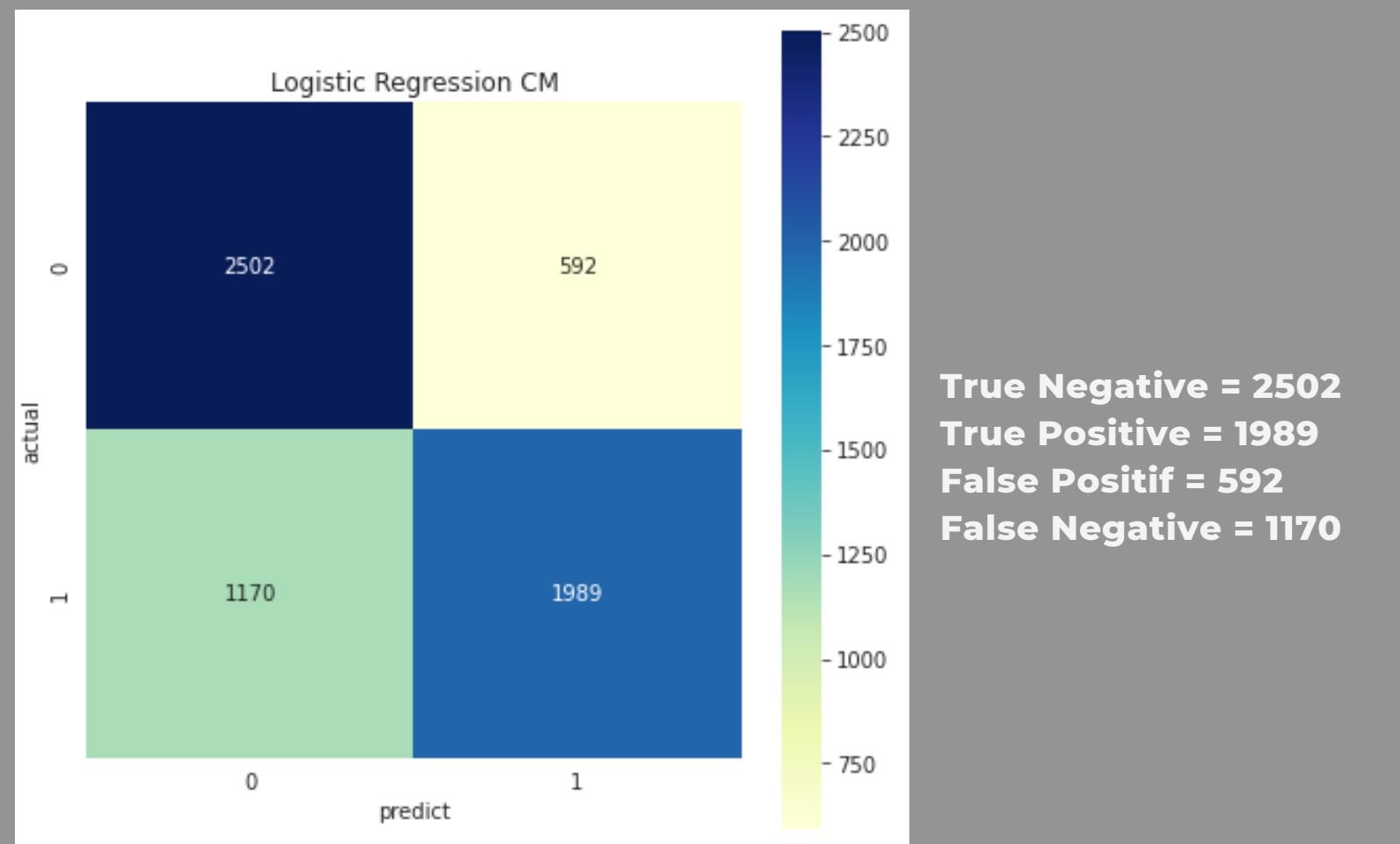
Classification Report

	precision	recall	f1-score	support
0	0.68	0.81	0.74	3094
1	0.77	0.63	0.69	3159
accuracy			0.72	6253
macro avg	0.73	0.72	0.72	6253
weighted avg	0.73	0.72	0.72	6253

Sensitivity score : 63 %
specificity score : 81 %

Accuracy: 0.7182152566767951
Accuracy: 72 %

Confusion Matrix



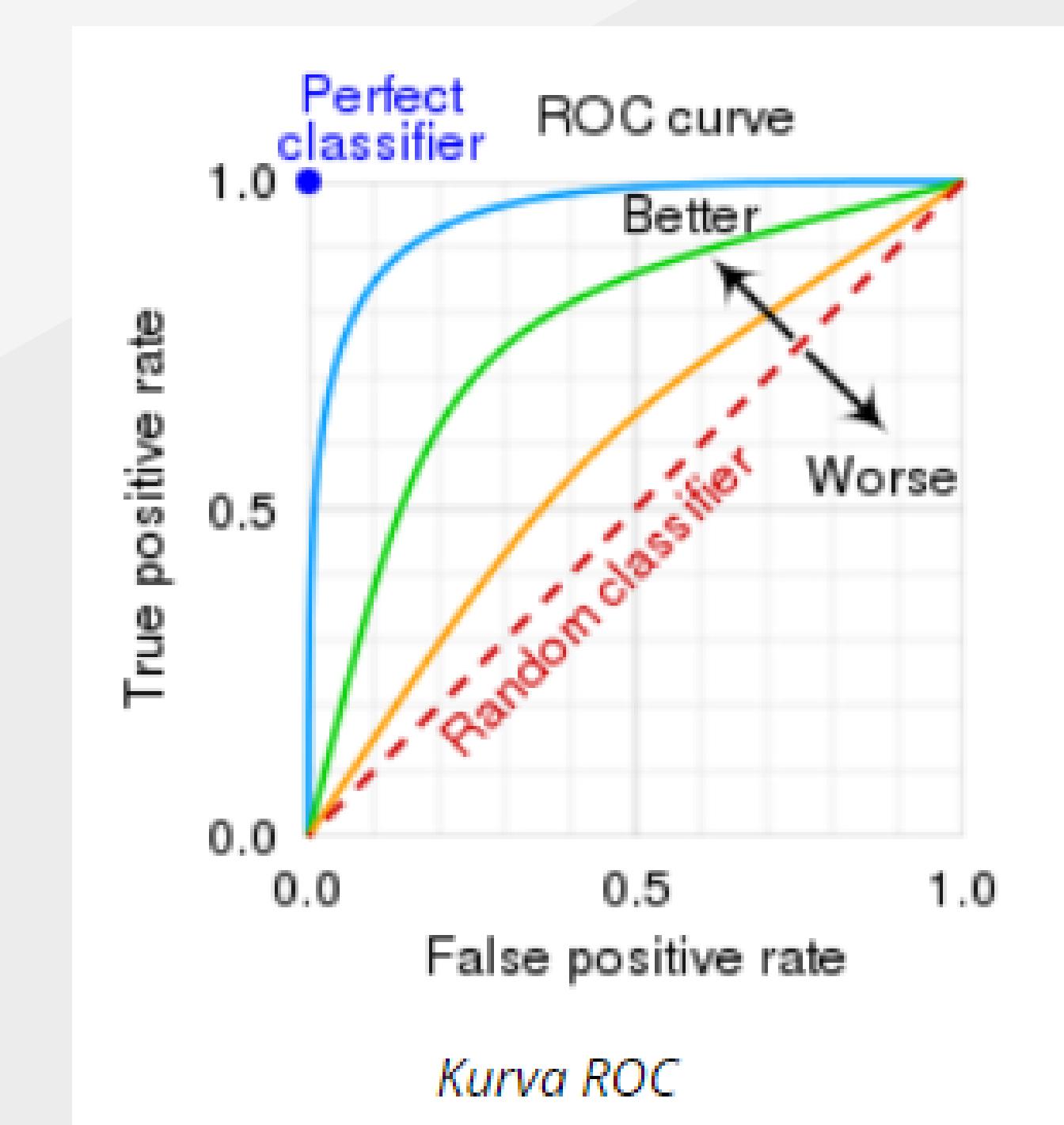
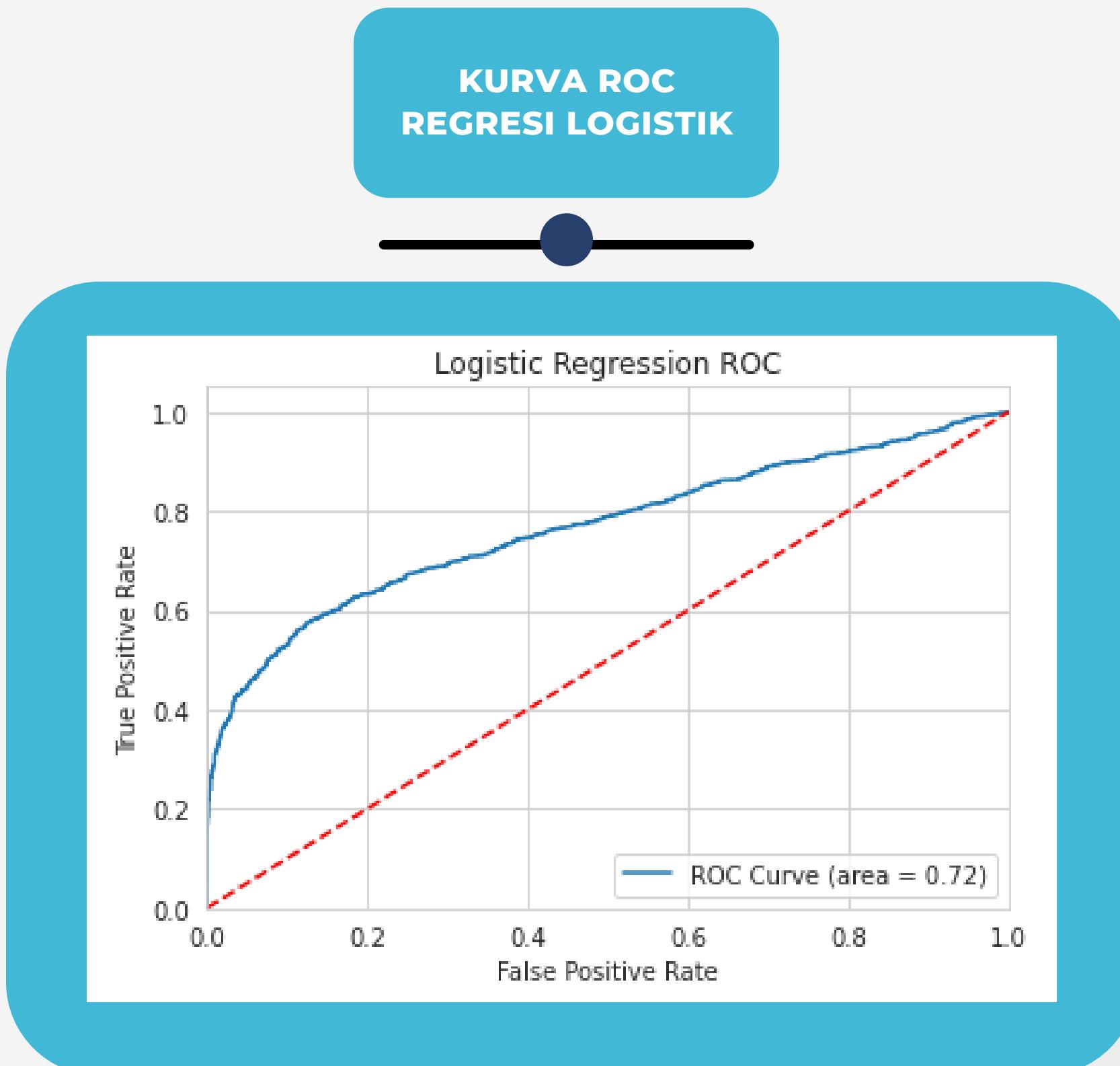
MODEL EVALUATION

Hasil:

1. Akurasi model yang didapatkan ialah sebesar 72%
2. Sensitivity (recall 1) : perbandingan antara True Positive (TP) dengan banyaknya data yang sebenarnya positif sebesar 63%
3. Specificity (recall 0) : perbandingan antara True Negatif (TN) dengan banyaknya data yang sebenarnya negatif sebesar 81%
4. precision : perbandingan antara True Positive (TP) dengan banyaknya data yang diprediksi positif sebesar 77% dan perbandingan antara True Negatif (TN) dengan banyaknya data yang diprediksi negatif sebesar 68%
5. F1-Score : perbandingan rata-rata precision dan recall sebesar 0,74 untuk kelas 0 dan 0,69 untuk kelas 1. Nilai terbaik F1-Score adalah 1.0 dan nilai terburuknya adalah 0. Secara representasi, jika F1-Score punya skor yang baik mengindikasikan bahwa model klasifikasi punya precision dan recall yang baik.

PERBANDINGAN ROC CURVE

Kurva ROC menunjukkan visualisasi antara true positive rate (TPR) dan false positive rate (FPR). Classifier yang memberikan kurva semakin mendekat ke sudut kiri atas (perfect classifier) menunjukkan kinerja yang semakin baik.Semakin dekat kurva ke titik-titik yang terletak di sepanjang diagonal ($FPR = TPR$) yaitu diagonal 45 derajat dari ruang ROC, maka semakin tidak akurat classifier tersebut.



MODEL SELECTION

RANDOM FOREST

Random forest adalah algoritma supervised machine learning yang merupakan sebuah model ensemble, yaitu model yang dibentuk dari banyak model Decision Tree, baik untuk regresi maupun untuk klasifikasi, dengan menerapkan metode bootstrap aggregating (bagging) dan random feature selection

Model Implementation

mengimplementasikan data kedalam model machine learning , dengan melatih model dan melakukan prediksi

Model Evaluation

melakukan evaluasi terhadap model dan hasil prediksinya dengan melakukan uji statistik

MODEL IMPLEMENTATION

Mendefinisikan X dan Y yang baru yang mana telah dilakukan resampling dan feature selection kedalam bentuk train test split data

```
"train_test_split(smote_X, smote_y, test_size=0.2, random_state=0)"
```

Membuat classifiernya

```
"clf=RandomForestClassifier(n_estimators=100)"
```

Latih model menggunakan data pelatihan

```
"clf.fit(X_train,y_train)"
```

Prediksi model menggunakan data test

```
"y_pred = clf.predict(X_test)"
```

```
[970] # membagi data yang telah disiapkan untuk model kedalam data train test split
      X_train, X_test, y_train, y_test = train_test_split(smote_X,smote_y, test_size=0.2,random_state=0)

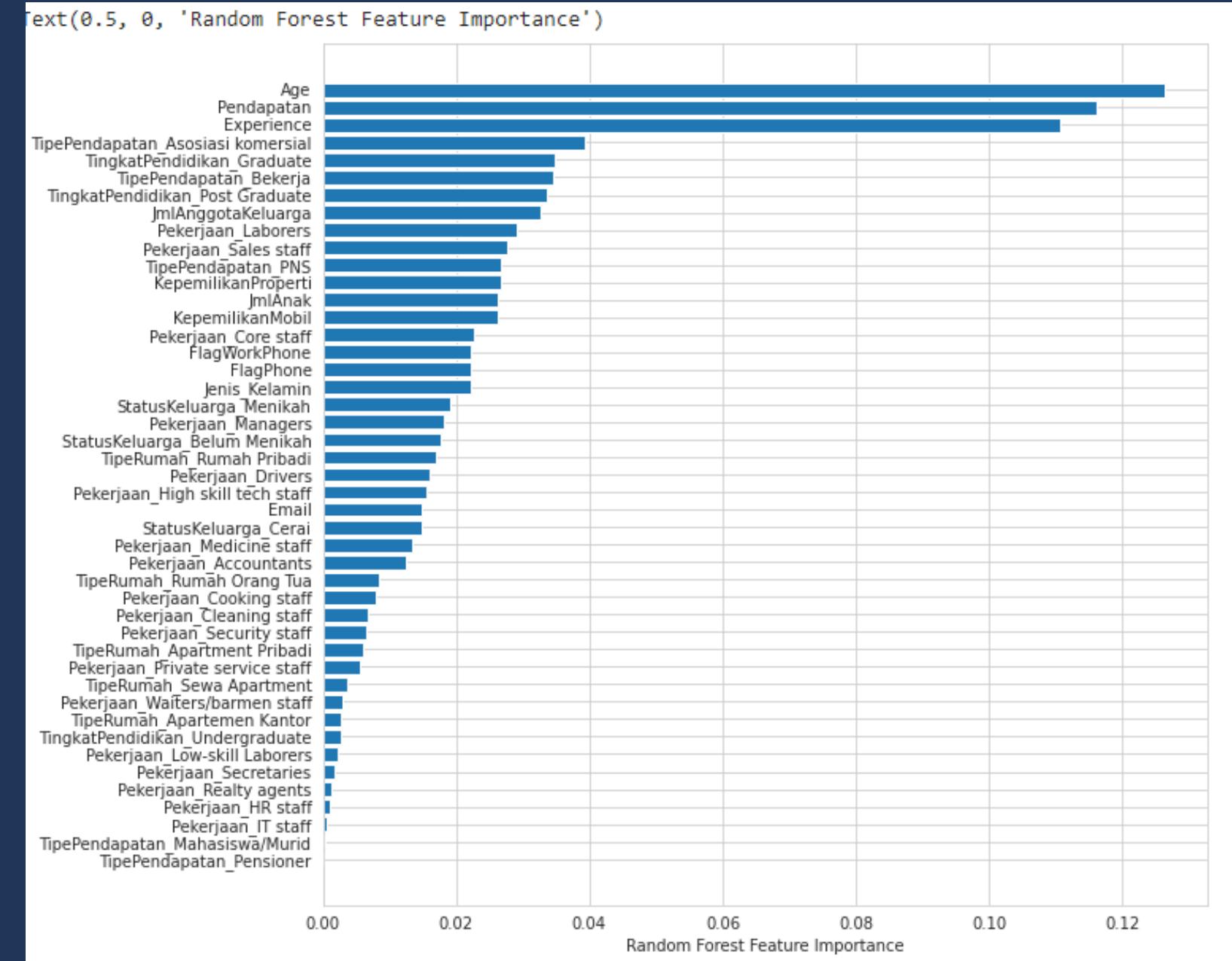
      #Import Random Forest Model
      from sklearn.ensemble import RandomForestClassifier

      #buat Gaussian Classifiernya
      clf=RandomForestClassifier(n_estimators=100)

      #latih model menggunakan data pelatihan
      clf.fit(X_train,y_train)

      RandomForestClassifier()

[972] # prediksi model menggunakan data test
      y_pred=clf.predict(X_test)
```



FEATURE IMPORTANCE

menampilkan feature importance pada model random forest

Random forest merupakan ensemble learning metode di mana beberapa algoritma pembelajaran digunakan secara bersamaan, lalu dikombinasikan untuk mendapatkan hasil pemodelan yang lebih akurat. secara umum memberikan kinerja prediktif yang baik, overfitting rendah, dan interpretasi yang mudah. Pemilihan fitur dalam random forest termasuk dalam kategori Embedded methods. Mereka diimplementasikan oleh algoritme yang memiliki metode pemilihan fitur bawaannya sendiri dengan melakukan pengujian fitur pada pohon keputusan . tidak seperti regresi yang pemilihan fiturnya dilakukan dengan metode terpisah.

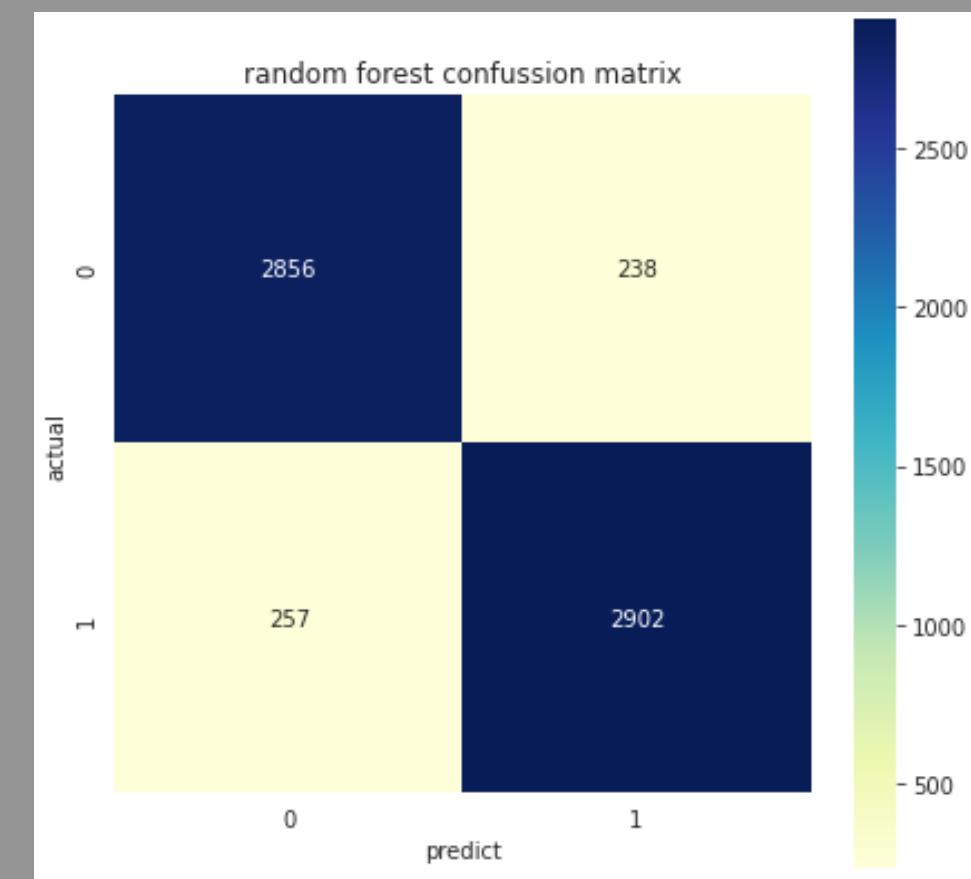
Classification Report

	precision	recall	f1-score	support
0	0.92	0.92	0.92	3094
1	0.92	0.92	0.92	3159
accuracy			0.92	6253
macro avg	0.92	0.92	0.92	6253
weighted avg	0.92	0.92	0.92	6253

Accuracy: 0.9208379977610747
accuracy: 92 %

Sensitivity score : 92 %
specificity score : 92 %

Confusion Matrix



OOOO

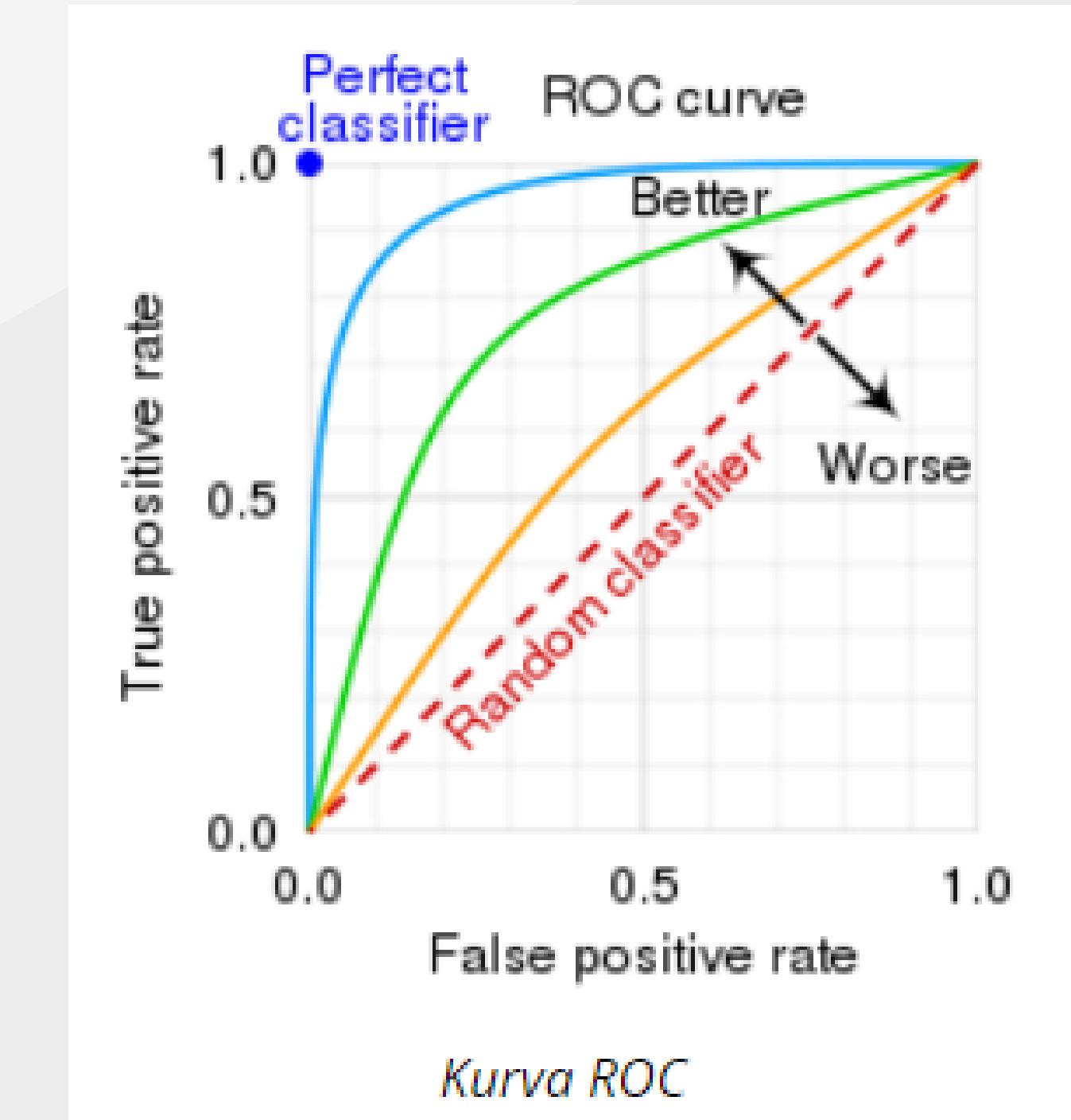
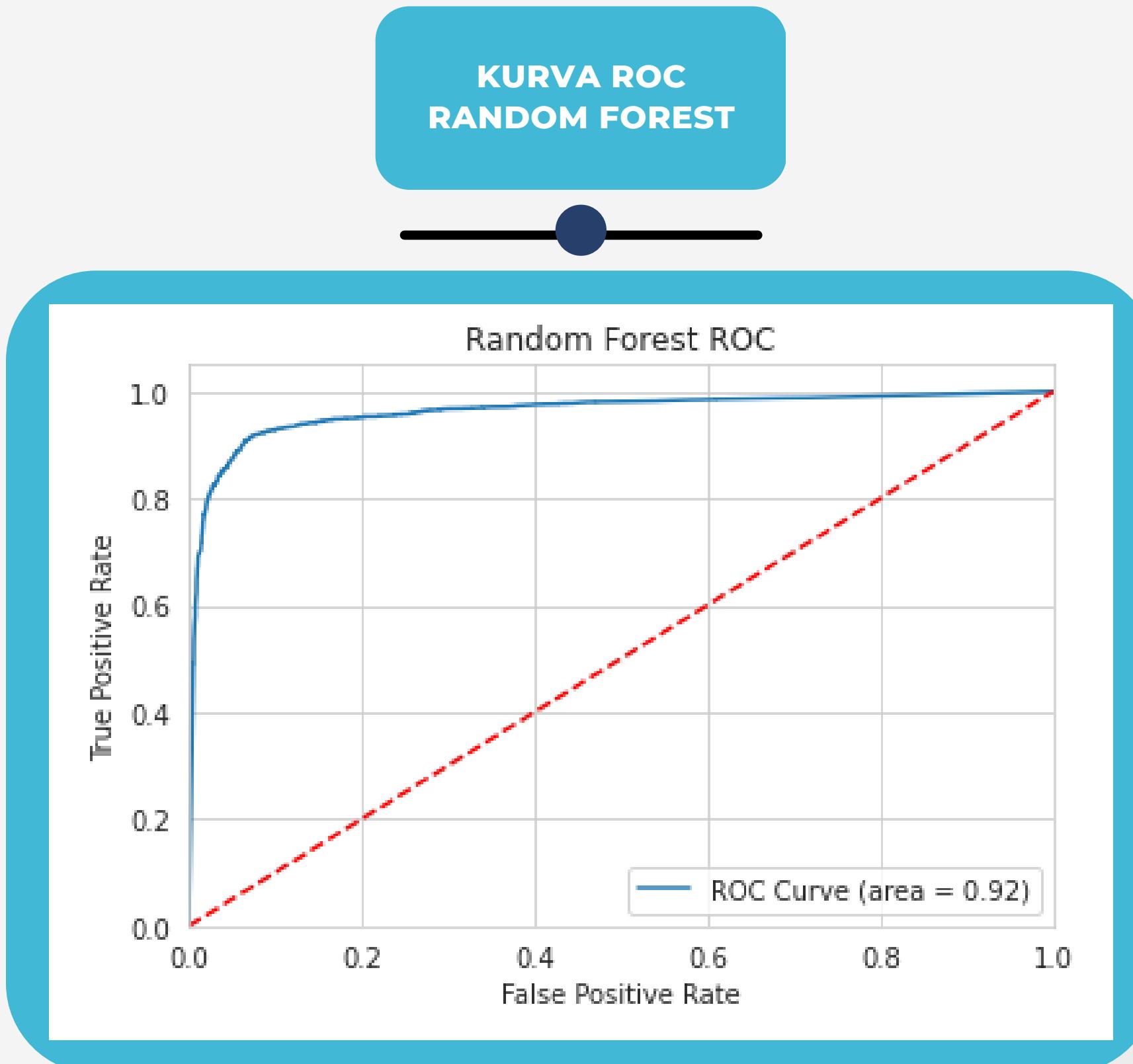
MODEL EVALUATION

Hasil:

1. Akurasi model yang didapatkan ialah sebesar 92%
2. Sensitivity (recall 1) : perbandingan antara True Positive (TP) dengan banyaknya data yang sebenarnya positif sebesar 92%
3. Specificity (recall 0) : perbandingan antara True Negatif (TN) dengan banyaknya data yang sebenarnya negatif sebesar 92%
4. precision : perbandingan antara True Positive (TP) dengan banyaknya data yang diprediksi positif sebesar 92% dan perbandingan antara True Negatif (TN) dengan banyaknya data yang diprediksi negatif sebesar 92%
5. F1-Score : perbandingan rata-rata precision dan recall sebesar 0,92 untuk kelas 0 dan 0,92 untuk kelas 1. Nilai terbaik F1-Score adalah 1,0 dan nilai terburuknya adalah 0. Secara representasi, jika F1-Score punya skor yang baik mengindikasikan bahwa model klasifikasi punya precision dan recall yang baik.

PERBANDINGAN ROC CURVE

Kurva ROC menunjukkan visualisasi antara true positive rate (TPR) dan false positive rate (FPR). Classifier yang memberikan kurva semakin mendekat ke sudut kiri atas (perfect classifier) menunjukkan kinerja yang semakin baik. Semakin dekat kurva ke titik-titik yang terletak di sepanjang diagonal ($FPR = TPR$) yaitu diagonal 45 derajat dari ruang ROC, maka semakin tidak akurat classifier tersebut.



06 DEPLOYMENT MODEL

```
# Menampilkan dataframe
df_test

   Id_customer  JK KepemilikanMobil KepemilikanProperti JmlAnak Pendapatan TingkatPendidikan StatusKeluarga TipeRumah FlagWorkPhone FlagPhone Email Pekerjaan JmlAnggotaKeluarga Age Experience
0    5142248 Perempuan      False        True     0   225000          PNS       Graduate     Menikah    Rumah pribadi    False  False  False  Private service staff  2   54      6
1    5036925 Perempuan      True        True     0   157500  Asosiasi komersial  Graduate     Menikah    Rumah pribadi    True   True  True  Core staff  2   33      8
2    5126080 Perempuan      False        True     1   112500          PNS       Graduate     Menikah    Rumah pribadi    False  False  False  Managers  3   41      7
3    5088887 Perempuan      False        True     0   171000        Bekerja  Graduate Belum Menikah  Sewa Apartemen  False  False  False  Laborers  1   46      2
4    5022156 Perempuan      True        True     2   180000  Asosiasi komersial Postgraduate     Menikah    Rumah pribadi    False  True  True  NaN  4   32      8
```

```
pred=clf.predict(new_df_test)
print(pred)

[0 0 0 0 0 1 1 1 0 1 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0
 1 0 1 0 0 1 1 1 1 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 1 0 1 0 1 0 1 1 0
 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 0 0 1 0 0 0 1 1 0 1 1 0
 0 0 1 1 0 0 0 1 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0
 0 0 1 1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 1
 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 1]
```

```
✓ [417] # menampilkan Id_customer dan hasil prediksinya
df_test

   Id_customer  Overdue_Prediction
0    5142248            0
1    5036925            0
2    5126080            0
3    5088887            0
4    5022156            0
...
195   5105368            0
196   5116026            1
197   5067627            0
198   5090052            0
199   5023668            0
200 rows x 2 columns
```

```
✓ [421] # menghitung jumlah tiap values dari hasil prediksi
df_test['Overdue_Prediction'].value_counts()

0    128
1     72
Name: Overdue_Prediction, dtype: int64
```

read data test

preprocessing data test

deploy model

predict data test



KESIMPULAN

- model berhasil memprediksi data dengan baik khususnya pada algoritma random forest yang memiliki classification report lebih baik dibandingkan dengan regresi logistik ,yaitu dengan memiliki nilai akurasi dan F1-score yang jauh lebih besar serta ditunjukan dengan nilai kurva ROC yang tinggi ini menunjukkan bahwa model memiliki kinerja yang bagus yang dapat digunakan untuk menyelesaikan masalah klasifikasi pada dataset.
- algoritma regresi logistik tidak cocok untuk melakukan klasifikasi pada dataset ini karena model yang terlalu sederhana dalam mempelajari data (underfitting)



