# Tugas Pendahuluan

**Tugas Pendahuluan dikerjakan dengan dataset titanic yang dapat didownload pada link berikut. Lakukan prediksi apakah suatu penumpang selamat atau tidak (kolom survived), bernilai 0 jika tidak selamat, dan 1 jika selamat.**

**Tugas dikerjakan secara berkelompok, dengan 1 kelompok terdiri atas 2 mahasiswa. Waktu pengerjaan dari 28 Maret 2022 - 3 April 2022 pukul 23.59.**

# 0. Loading Data and Library

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import model_selection
import sklearn.metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import GridSearchCV, cross_validate
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn import svm
from sklearn.ensemble import VotingClassifier, StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import MinMaxScaler
```

In [2]:

```
df = pd.read_csv("titanic_dataset.csv")
df
```

Out[2]:

| | index | pclass | survived | name | sex | age | sibsp | parch | ticket | fare | cabin | embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3.0 | 1.0 | Abelseth, Miss. Karen Marie | female | 16.0 | 0.0 | 0.0 | 348125 | 7.6500 | NaN | S |
| **1** | 1 | 3.0 | 0.0 | Burns, Miss. Mary Delia | female | 18.0 | 0.0 | 0.0 | 330963 | 7.8792 | NaN | Q |
| **2** | 2 | 1.0 | 1.0 | Fortune, Miss. Alice Elizabeth | female | 24.0 | 3.0 | 2.0 | 19950 | 263.0000 | C23 C25 C27 | S |
| **3** | 3 | 3.0 | 1.0 | de Messemaeker, Mrs. Guillaume Joseph (Emma) | female | 36.0 | 1.0 | 0.0 | 345572 | 17.4000 | NaN | S |
| **4** | 4 | 3.0 | 0.0 | Jonsson, Mr. Nils Hilding | male | 27.0 | 0.0 | 0.0 | 350408 | 7.8542 | NaN | S |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1304** | 1304 | 3.0 | 1.0 | Dahl, Mr. Karl Edwart | male | 45.0 | 0.0 | 0.0 | 7598 | 8.0500 | NaN | S |
| **1305** | 1305 | 1.0 | 0.0 | Penasco y Castellana, Mr. Victor de Satode | male | 18.0 | 1.0 | 0.0 | PC 17758 | 108.9000 | C65 | C |
| **1306** | 1306 | 2.0 | 1.0 | Becker, Miss. Ruth Elizabeth | female | 12.0 | 2.0 | 1.0 | 230136 | 39.0000 | F4 | S |
| **1307** | 1307 | 3.0 | 1.0 | Murphy, Miss. Katherine "Kate" | female | NaN | 1.0 | 0.0 | 367230 | 15.5000 | NaN | Q |
| **1308** | 1308 | 3.0 | 0.0 | Sage, Mr. Frederick | male | NaN | 8.0 | 2.0 | CA. 2343 | 69.5500 | NaN | S |

1309 rows × 12 columns

# I. Data Understanding

**Tujuan dari bagian ini adalah peserta dapat memahami kualitas dari data yang diberikan. Hal ini meliputi:**

1. **Ukuran data**
2. **Statistik dari tiap fitur**
3. **Pencilan (outlier)**
4. **Korelasi**
5. **Distribusi**

## I.1

**Carilah:**

1. **Ukuran dari data (instances dan features)**
2. **Tipe dari tiap-tiap fitur**
3. **Banyaknya unique values dari fitur yang bertipe kategorikal**
4. **Nilai minimum, maksimum, rata-rata, median, dan standar deviasi dari fitur yang tidak bertipe kategorikal**

In [12]:

```
instance = len(df)
features = len(df.columns)
feature_type = df.dtypes
df_categ = df[['name', 'sex', 'ticket', 'cabin', 'embarked']].copy()
df_noncateg = df[['pclass', 'survived', 'age', 'sibsp', 'parch', 'fare']].copy()
# uniqueValues = (df['name'].append(df['sex']).append(df['ticket']).append(df['cabin']).a
ppend(df['embarked'])).nunique()
uniqueValues2 = df_categ.nunique()
mini = df_noncateg.min()
maks = df_noncateg.max()
rata_rata = df_noncateg.mean()
med = df_noncateg.median()
std_dev = df_noncateg.std()

print("1. Ukuran dari data: instances = " + str(instance) + " , feature = " + str(featur
es))
print()
print("2. Tipe dari tiap-tiap fitur: ")
print(feature_type)
print()
print("3. Banyaknya unique values dari fitur yang bertipe kategorikal: ")
# print(uniqueValues)
print(uniqueValues2)
```

```
1. Ukuran dari data: instances = 1309 , feature = 12

2. Tipe dari tiap-tiap fitur:
index          int64
pclass       float64
survived     float64
name          object
sex           object
age          float64
sibsp        float64
parch        float64
ticket        object
fare         float64
cabin         object
embarked      object
dtype: object

3. Banyaknya unique values dari fitur yang bertipe kategorikal:
name        1307
sex            2
ticket       929
cabin        186
```

```
embarked       3
dtype: int64
```

In [13]:

```
print("4. Nilai minimum, maksimum, rata-rata, median, dan standar deviasi dari fitur yang tidak bertipe kategorikal: ")
print()
print("Nilai Minimum: ")
print(mini)
print()
print("Nilai Maksimum: ")
print(maks)
print()
print("Nilai Rata-Rata: ")
print(rata_rata)
```

```
4. Nilai minimum, maksimum, rata-rata, median, dan standar deviasi dari fitur yang tidak bertipe kategorikal: 

Nilai Minimum: 
pclass      1.0000
survived    0.0000
age         0.1667
sibsp       0.0000
parch       0.0000
fare        0.0000
dtype: float64

Nilai Maksimum: 
pclass        3.0000
survived      1.0000
age          80.0000
sibsp         8.0000
parch         9.0000
fare        512.3292
dtype: float64

Nilai Rata-Rata: 
pclass       2.294882
survived     0.381971
age         29.881135
sibsp        0.498854
parch        0.385027
fare        33.295479
dtype: float64
```

In [14]:

```
print("Nilai Median: ")
print(med)
print()
print("Nilai Standar Deviasi: ")
print(std_dev)
```

```
Nilai Median: 
pclass       3.0000
survived     0.0000
age         28.0000
sibsp        0.0000
parch        0.0000
fare        14.4542
dtype: float64

Nilai Standar Deviasi: 
pclass       0.837836
survived     0.486055
age         14.413500
sibsp        1.041658
parch        0.865560
fare        51.758668
dtype: float64
```

```
dtype: float64
```

## I.2

**Carilah:**

1. **Missing values dari tiap fitur**
2. **Outliers dari tiap fitur (gunakan metode yang kalian ketahui)**

In [15]:

```
miss_val = df.isnull().sum()
print("1. Missing values dari tiap fitur: ")
print(miss_val)
```

```
1. Missing values dari tiap fitur:
index          0
pclass         0
survived       0
name           0
sex            0
age          263
sibsp          0
parch          0
ticket         0
fare           1
cabin       1014
embarked       2
dtype: int64
```

In [16]:

```
def outlier(column):
    Q1 = np.percentile(df[column], 25,
                    interpolation = 'midpoint')

    Q3 = np.percentile(df[column], 75,
                   interpolation = 'midpoint')

    IQR = Q3 - Q1

    upper = df[column] >= (Q3+1.5*IQR)
    count_upper = len(np.where(upper)[0])

    lower = df[column] <= (Q1-1.5*IQR)
    count_lower = len(np.where(lower)[0])

    count_outlier = count_upper + count_lower
    return(count_outlier)
    # print(np.where(upper))
    # print(np.where(lower))

print("2. Outliers dari tiap fitur: ")

for col in df_noncateg.columns:
    print(col + ": " + str(outlier(col)))
    # print(col + ": ")
    # outlier(col)
# outlier('pclass')
```

```
2. Outliers dari tiap fitur:
pclass: 0
survived: 0
age: 0
sibsp: 57
parch: 2311
fare: 0
```

## I.3

**Carilah:**

1. **Korelasi antar fitur**
2. **Visualisasikan distribusi dari tiap fitur (kategorikal dan kontinu)**
3. **Visualisasikan distribusi dari tiap fitur, dengan data dibagi tiap unique values fitur survived**

In [17]:

```
print("1. Korelasi antar fitur: ")
print(df.corr())
```

```
1. Korelasi antar fitur:
             index    pclass  survived       age     sibsp     parch      fare
index     1.000000 -0.003466  0.002967 -0.003376 -0.015501 -0.013538 -0.022424
pclass   -0.003466  1.000000 -0.312469 -0.408106  0.060832  0.018322 -0.558629
survived  0.002967 -0.312469  1.000000 -0.055513 -0.027825  0.082660  0.244265
age      -0.003376 -0.408106 -0.055513  1.000000 -0.243699 -0.150917  0.178739
sibsp    -0.015501  0.060832 -0.027825 -0.243699  1.000000  0.373587  0.160238
parch    -0.013538  0.018322  0.082660 -0.150917  0.373587  1.000000  0.221539
fare     -0.022424 -0.558629  0.244265  0.178739  0.160238  0.221539  1.000000
```

In [157]:

```
print("2. Visualisasi distribusi dari tiap fitur (kategorikal dan kontinu): ")
```

```
2. Visualisasi distribusi dari tiap fitur (kategorikal dan kontinu):
```

In [ ]:

```
print("index : ")
df['index'].plot.box()
```

In [ ]:

```
print("pclass : ")
df['pclass'].plot.box()
```

In [ ]:

```
print("survived : ")
df['survived'].plot.box()
```

In [ ]:

```
print("name : ")
GroupedData=df.groupby(by="name").size()
print(GroupedData.plot.box())
```

In [ ]:

```
print("sex : ")
GroupedData=df.groupby(by="sex").size()
print(GroupedData.plot.box())
```

In [ ]:

```
print("age : ")
df['age'].plot.box()
```

In [ ]:

```
print("sibsp : ")
df['sibsp'].plot.box()
```

In [ ]:

```
print("parch : ")
df['parch'].plot.box()
```

In [ ]:

```
print("ticket : ")
GroupedData=df.groupby(by="ticket").size()
print(GroupedData.plot.box())
```

In [ ]:

```
print("fare : ")
df['fare'].plot.box()
```

In [ ]:

```
print("cabin : ")
GroupedData=df.groupby(by="cabin").size()
print(GroupedData.plot.box())
```

In [ ]:

```
print("embarked : ")
GroupedData=df.groupby(by="embarked").size()
print(GroupedData.plot.box())
```

In [ ]:

```
print("3. Visualisasi distribusi dari tiap fitur, dengan data dibagi tiap unique values
fitur survived")
```

In [ ]:

```
print("index : ")
df.boxplot(by ='survived', column =['index'], grid = False)
```

In [ ]:

```
print("pclass : ")
df.boxplot(by ='survived', column =['pclass'], grid = False)
```

In [ ]:

```
print("name : ")
df.boxplot(by ='name', column =['survived'], grid = False)
```

In [ ]:

```
print("sex : ")
df.boxplot(by ='sex', column =['survived'], grid = False)
```

In [ ]:

```
print("age : ")
df.boxplot(by ='survived', column =['age'], grid = False)
```

In [ ]:

```
print("sibsp : ")
df.boxplot(by ='survived', column =['sibsp'], grid = False)
```

In [ ]:

```
print("parch : ")
df.boxplot(by ='survived', column =['parch'], grid = False)
```

In [ ]:

```
print("ticket : ")
df.boxplot(by ='ticket', column =['survived'], grid = False)
```

In [ ]:

```
print("fare : ")
df.boxplot(by ='survived', column =['fare'], grid = False)
```

In [ ]:

```
print("cabin : ")
df.boxplot(by ='cabin', column =['survived'], grid = False)
```

In [ ]:

```
print("embarked : ")
df.boxplot(by ='embarked', column =['survived'], grid = False)
```

# I.4

**Lakukanlah analisa pada data lebih lanjut jika dibutuhkan, kemudian lakukanlah:**

1. **Penambahan fitur jika memungkinkan**
2. **Pembuangan fitur yang menurut kalian tidak dibutuhkan**
3. **Penanganan missing values**
4. **Transformasi data kategorikal menjadi numerikal (encoding), dengan metode yang kalian inginkan**
5. **Lakukan scaling dengan MinMaxScaler**

In [19]:

```
print("1. Penambahan fitur jika memungkinkan")
```

In [ ]:

```
print("2. Pembuangan fitur yang menurut kalian tidak dibutuhkan")
```

In [ ]:

```
print("3. Penanganan missing values")
print("Penanganan Missing Value akan dilakukan dengan metode 'Filling null values with the previous ones'")
print("dan 'Filling null value with the next ones'")
print("Kedua metode ini mengisi missing value dengan nilai dari instance sebelumnya dan instance sesudahnya")
# Mengisi missing value dengan nilai dari instance sebelumnya kemudian dengan nilai instance sesudahnya
print(df.fillna(method ='pad').fillna(method ='bfill').isna().sum())
```

In [ ]:

```
print("4. Transformasi data kategorikal menjadi numerikal (encoding), dengan metode yang kalian inginkan")
print("Metode transformasi yang kami gunakan adalah dengan mengganti setiap unique value dari feature kategorikal")
print("menjadi angka numerik yang membedakan setiap unique value")
# pd.get_dummies(df, prefix=None, prefix_sep='_', dummy_na=False, columns=None, sparse=False, drop_first=False, dtype=None)
arr_uniq_name = []
arr_uniq_sex = []
arr_uniq_ticket = []
arr_uniq_cabin = []
arr_uniq_embarked = []

for i in range(1, df['name'].nunique() + 1):
    arr_uniq_name.append(i)
for i in range(1, df['sex'].nunique() + 1):
    arr_uniq_sex.append(i)
for i in range(1, df['ticket'].nunique() + 1):
    arr_uniq_ticket.append(i)
for i in range(1, df['cabin'].nunique() + 1):
    arr_uniq_cabin.append(i)
```

```
for i in range(1, df['embarked'].nunique() + 1):
    arr_uniq_embarked.append(i)

df['name'].replace(df['name'].unique(),
                        arr_uniq_name, inplace=True)
df['sex'].replace(df['sex'].unique(),
                        arr_uniq_sex, inplace=True)
df['ticket'].replace(df['ticket'].unique(),
                        arr_uniq_ticket, inplace=True)
df['cabin'].replace(df['cabin'].unique(),
                        arr_uniq_cabin, inplace=True)
df['embarked'].replace(df['embarked'].unique(),
                        arr_uniq_embarked, inplace=True)

print(df['name'])
print(df['sex'])
print(df['ticket'])
print(df['cabin'])
print(df['embarked'])
```

In [ ]:

```
print("5. Lakukan scaling dengan MinMaxScaler")
scaler = MinMaxScaler()

df_scaled = scaler.fit_transform(df.to_numpy())
df_scaled = pd.DataFrame(df_scaled, columns=[
  'index', 'pclass', 'survived', 'name', 'sex', 'age', 'sibsp', 'parch', 'ticket', 'fare
', 'cabin', 'embarked'])

print("Scaled Dataset Using MinMaxScaler")
df_scaled.head()
```

# II. Experiments Design

**Tujuan dari bagian ini adalah peserta dapat memahami cara melakukan eksperimen mencari metode terbaik dengan benar. Hal ini meliputi:**

1. **Pembuatan model**
2. **Proses validasi**
3. **Hyperparameter tuning**

## II.1

**Tentukanlah metrics yang akan digunakan pada eksperimen kali ini (dapat lebih dari 1 metric)**

1. **Precision**
2. **Recall**
3. **F1-Score**

## II.2

**Bagi data dengan perbandingan 0.8 untuk data train dan 0.2 untuk data validasi**

In [3]:

```
# II.2 Put your code here
#map male and female
df.sex = df.sex.map({'male':0,'female':1})

df.sex
```

Out[3]:

```
0        1
```

```
0        1
1        1
2        1
3        1
4        0
        ..
1304     0
1305     0
1306     1
1307     1
1308     0
Name: sex, Length: 1309, dtype: int64
```

In [4]:

```
#isi null di age
df['age'].fillna(df['age'].mean(), inplace= True)
df['fare'].fillna(df['fare'].mean(), inplace= True)
df.drop(['cabin','embarked'],axis=1,inplace=True)

y = df.survived.copy()
x = df.drop(['survived'], axis=1)
x.drop(['name','ticket','index'],axis=1,inplace=True)

x_train , x_val , y_train , y_val = train_test_split(x,y,test_size=0.2,random_state=1)
```

In [5]:

```
np.any(np.isnan(x_train))
```

Out[5]:

```
False
```

# II.3

**Lakukanlah:**

1. **Prediksi dengan menggunakan model Logistic Regression sebagai *baseline***
2. **Tampilkan evaluasi dari model yang dibangun dari metrics yang anda tentukan pada II.1**
3. **Tampilkan confusion matrix**

In [6]:

```
# II.3 Put your code here
model = LogisticRegression()

#fit
model.fit(x_train, y_train)
model.predict(x_val)

print("================Evaluasi================")
print(classification_report(y_val,model.predict(x_val)))

print("================Confusion Matrix================")
print(confusion_matrix(y_val,model.predict(x_val)))
```

```
================Evaluasi================
              precision    recall  f1-score   support

         0.0       0.84      0.85      0.84       157
         1.0       0.77      0.75      0.76       105

    accuracy                           0.81       262
   macro avg       0.80      0.80      0.80       262
weighted avg       0.81      0.81      0.81       262

================Confusion Matrix================
[[133  24]
 [ 26  79]]
```

## II.4

**Lakukanlah:**

1. **Pembelajaran dengan model lain**
2. **Hyperparameter tuning model yang kalian pakai dengan menggunakan Grid Search (perhatikan random factor pada beberapa algoritma model)**
3. **Lakukan validasi dengan menggunakan cross validation**

In [24]:

```
# # II.4 Put your code here
# # Menggunakan SVM
# # Grid Search
modelSVM = GridSearchCV(svm.SVC(gamma='auto'),{
    'C' : [1,10,20],
    'kernel' : ['rbf','linear']
}, cv = 5, return_train_score=False)

modelSVM.fit(x_train,y_train)


# 5 Fold Cross Validation
labels = ['Logistic Regression' , 'SVM']

for model_ , label in zip([model,modelSVM],labels):
    scores1 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'precision')
    scores2 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'recall')
    scores3 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'f1')
    print("Precision: %0.2f (+/- %0.2f) [%s]"
        % (scores1.mean(), scores1.std(),label))
    print("Recall: %0.2f (+/- %0.2f) [%s]"
        % (scores2.mean(), scores2.std(),label))
    print("F1 : %0.2f (+/- %0.2f) [%s]"
        % (scores3.mean(), scores3.std(),label))
```

# III. Improvement

**Terdapat beberapa metode untuk melakukan peningkatan performa, contohnya adalah:**

1. **Melakukan oversampling / undersampling pada data**
2. **Menggabungkan beberapa model**

**Pada bagian ini, kalian diharapkan dapat:**

1. **Melakukan training dengan data hasil oversampling / undersampling dan melakukan validasi dengan benar**
2. **Memahami beberapa metode untuk menggabungkan beberapa model**

## III.1

**Lakukanlah:**

1. **Oversampling pada kelas minoritas pada data train, kemudian train dengan model *baseline* (II.3), lakukan validasi dengan data validasi. Data train dan validasi adalah data yang kalian bagi pada bagian II.2**
2. **Undersampling pada kelas mayoritas pada data train, kemudian train dengan model *baseline* (II.3) lakukan validasi dengan data validasi. Data train dan validasi adalah data yang kalian bagi pada bagian II.2**

In [7]:

```
%pip install imblearn
```

```
Requirement already satisfied: imblearn in c:\users\fahkry\anaconda3\lib\site-packages (0
 0)
```

```
.0)
Requirement already satisfied: imbalanced-learn in c:\users\fahkry\anaconda3\lib\site-pac
kages (from imblearn) (0.9.0)
Requirement already satisfied: scipy>=1.1.0 in c:\users\fahkry\anaconda3\lib\site-package
s (from imbalanced-learn->imblearn) (1.7.3)
Requirement already satisfied: numpy>=1.14.6 in c:\users\fahkry\anaconda3\lib\site-packag
es (from imbalanced-learn->imblearn) (1.21.5)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\fahkry\anaconda3\lib\site
-packages (from imbalanced-learn->imblearn) (2.2.0)
Requirement already satisfied: scikit-learn>=1.0.1 in c:\users\fahkry\anaconda3\lib\site-
packages (from imbalanced-learn->imblearn) (1.0.2)
Requirement already satisfied: joblib>=0.11 in c:\users\fahkry\anaconda3\lib\site-package
s (from imbalanced-learn->imblearn) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [8]:

```python
from numpy import mean
from sklearn.datasets import make_classification
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import RepeatedStratifiedKFold
from sklearn.tree import DecisionTreeClassifier
from imblearn.pipeline import Pipeline
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
# define pipeline
steps = [('over', RandomOverSampler()), ('model', model)]
pipeline = Pipeline(steps=steps)
# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(pipeline, x_train, y_train, scoring='f1_micro', cv=cv, n_jobs=-
1)
score = mean(scores)
print('F1 Score: %.3f' % score)
```

```
F1 Score: 0.774
```

In [9]:

```python
# define pipeline
steps = [('under', RandomUnderSampler()), ('model', model)]
pipeline = Pipeline(steps=steps)
# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(pipeline, x_train, y_train, scoring='f1_micro', cv=cv, n_jobs=-
1)
score = mean(scores)
print('F1 Score: %.3f' % score)
```

```
F1 Score: 0.771
```

# III.2

**Lakukanlah:**

1. **Eksplorasi soft voting, hard voting, dan stacking**
2. **Buatlah model Logistic Regression dan SVM (boleh menggunakan model dengan beberapa parameter yang berbeda)**
3. **Lakukanlah soft voting dari model-model yang sudah kalian buat pada poin 2**
4. **Lakukan hard voting dari model-model yang sudah kalian buat pada poin 2**
5. **Lakukanlah stacking dengan final classifier adalah Logistic Regression dari model-model yang sudah kalian buat pada poin 2**
6. **Lakukan validasi dengan metrics yang kalian tentukan untuk poin 3, 4, dan 5**

**Put your answer for section III.2 point 1 here**

In [26]:

```
# III.2 Put your code here
modelLR = LogisticRegression()
modelSVM2 = svm.SVC(kernel='rbf', C=30, gamma='auto')

labels = ['Logistic Regression', 'SVM']

voting_model_hard = VotingClassifier(estimators = [(labels[0],modelLR),(labels[1], model
SVM2)],voting='hard')

voting_model_soft = VotingClassifier(estimators = [(labels[0],modelLR),(labels[1], model
SVM2)],voting='soft')
```

In [27]:

```
labels_new = ['Logistic Regression', 'SVM', 'Voting Classifier Hard', 'Voting Classifier
Soft']
# Voting Soft dan Hard
for (model_ , label) in zip([modelLR,modelSVM2,voting_model_hard,voting_model_soft], lab
els_new):
    scores1 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'precision')
    scores2 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'recall')
    scores3 = model_selection.cross_val_score(model_,x,y,cv = 5, scoring = 'f1')
    print("Precision: %0.2f (+/- %0.2f) [%s]"
          % (scores1.mean(), scores1.std(),label))
    print("Recall: %0.2f (+/- %0.2f) [%s]"
          % (scores2.mean(), scores2.std(),label))
    print("F1 : %0.2f (+/- %0.2f) [%s]"
          % (scores3.mean(), scores3.std(),label))

#Stacking
estimators = [
    ('svm',modelSVM2)
]

stack = StackingClassifier(
    estimators=estimators,
    final_estimator=modelLR
)

result1 = cross_validate(estimator=stack, X=x, y = y, cv=10, scoring='precision')
result2 = cross_validate(estimator=stack, X=x, y = y, cv=10, scoring='recall')
result3 = cross_validate(estimator=stack, X=x, y = y, cv=10, scoring='f1')
label = 'Stack'
print("Precision: %0.2f (+/- %0.2f) [%s]"
          % (result1['test_score'].mean(), result1['test_score'].std(),label))
print("Recall: %0.2f (+/- %0.2f) [%s]"
          % (result2['test_score'].mean(), result2['test_score'].std(),label))
print("F1 : %0.2f (+/- %0.2f) [%s]"
          % (result3['test_score'].mean(), result3['test_score'].std(),label))
```

```
Precision: 0.74 (+/- 0.05) [Logistic Regression]
Recall: 0.68 (+/- 0.05) [Logistic Regression]
F1 : 0.71 (+/- 0.04) [Logistic Regression]
Precision: 0.61 (+/- 0.05) [SVM]
Recall: 0.58 (+/- 0.04) [SVM]
F1 : 0.60 (+/- 0.04) [SVM]
Precision: 0.76 (+/- 0.04) [Voting Classifier Hard]
Recall: 0.46 (+/- 0.04) [Voting Classifier Hard]
F1 : 0.57 (+/- 0.04) [Voting Classifier Hard]
```

```
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
```

```
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
```

```
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False
```

```
  warnings.warn(
C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py:696: Us
erWarning: Scoring failed. The score on this train-test partition for these parameters wi
ll be set to nan. Details:
Traceback (most recent call last):
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\model_selection\_validation.py
", line 687, in _score
    scores = scorer(estimator, X_test, y_test)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 87,
in __call__
    score = scorer._score(cached_call, estimator,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 236,
in _score
    y_pred = method_caller(estimator, "predict", X)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\metrics\_scorer.py", line 53,
in _cached_call
    return getattr(estimator, method)(*args, **kwargs)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 309
, in predict
    maj = np.argmax(self.predict_proba(X), axis=1)
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 329
, in _predict_proba
    avg = np.average(self._collect_probas(X), axis=0,
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in _collect_probas
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\ensemble\_voting.py", line 324
, in <listcomp>
    return np.asarray([clf.predict_proba(X) for clf in self.estimators_])
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 666, in pr
edict_proba
    self._check_proba()
  File "C:\Users\joseg\anaconda3\lib\site-packages\sklearn\svm\_base.py", line 633, in _c
heck_proba
    raise AttributeError("predict_proba is not available when "
AttributeError: predict_proba is not available when  probability=False

  warnings.warn(
```

```
Precision: nan (+/- nan) [Voting Classifier Soft]
Recall: nan (+/- nan) [Voting Classifier Soft]
F1 : nan (+/- nan) [Voting Classifier Soft]
```

# IV. Analisis

**Bandingkan hasil dari:**

1. **Model Baseline (II.3)**
2. **Model lain (II.4)**
3. **Hasil undersampling**
4. **Hasil oversampling**
5. **Hasil soft voting**
6. **Hasil hard voting**
7. **Hasil stacking**

**Model Baseline : Memiliki Presisi, F1, dan Recall yang cukup tinggi. Model mampu membuat prediksi True Positives </br> Model lain : Memiliki Presisi, F1, dan Recall yang cukup tinggi. Model mampu membuat prediksi True Positives</br> Undersampling : Memiliki F1 yang lebih rendah dari Oversampling, Model kurang mampu membuat prediksi True Positives </br> Oversampling : Memiliki F1 yang lebih tinggi dari Undersampling, Model mampu membuat prediksi True Positives </br> Soft Voting : Tidak memiliki nilai</br> Hard Voting : Memiliki Presisi, F1, dan Recall yang cukup rendah tapi masih lebih tinggi dari Stacking. Model kurang mampu membuat prediksi True Positives</br> Stacking : Memiliki Presisi, F1, dan Recall yang cukup rendah. Model kurang mampu membuat prediksi True Positives</br>**