

# ABSOLUTE

Kelompok 7:

1. Fikrie Lazuardi
2. Retno Dwi
3. Yunita Rachmawati
4. Rian Dwi Haryono
5. Wahyuni Salman
6. Retno Harindhi
7. Hidayat Yatul



The background of the slide is a faded, grayscale aerial photograph of a dense urban skyline, likely New York City, showing numerous skyscrapers and a complex network of roads and highways.

## STAGE 3

# Modelling

```
Accuracy (Test Set): 0.91
Precision (Test Set): 0.59
Recall (Test Set): 0.37
F1-Score (Test Set): 0.45
roc_auc (test-proba): 0.92
roc_auc (train-proba): 0.99
roc_auc (crossval train): 0.9765877507321999
roc_auc (crossval test): 0.5891293229797601
```

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
import numpy as np

#Menjadikan ke dalam bentuk dictionary
hyperparameters = {
    'max_depth' : [int(x) for x in np.linspace(10, 110, num = 11)],
    'min_child_weight' : [int(x) for x in np.linspace(1, 20, num = 11)],
    'gamma' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'tree_method' : ['auto', 'exact', 'approx', 'hist'],

    'colsample_bytree' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'eta' : [float(x) for x in np.linspace(0, 1, num = 100)],

    'lambda' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'alpha' : [float(x) for x in np.linspace(0, 1, num = 11)]
}

# Init
from xgboost import XGBClassifier
xg = XGBClassifier(random_state=42)
xg_tuned = RandomizedSearchCV(xg, hyperparameters, cv=5, random_state=42, scoring='recall')
xg_tuned.fit(X_train,Y_train)

# Predict & Evaluation
eval_classification(xg_tuned)
```

Ketika melakukan uji coba data dengan melakukan pemodelan, dapat disimpulkan bahwa diperoleh hasil nilai Recall tertinggi yaitu sebesar 0.34, lalu terdapat nilai Accuracy yaitu 0.85 dan didapatkan nilai ROC sebesar 1.00 pada saat menggunakan modeling Decision Tree Classifier.

# Modelling

## Random Forest

```
[ ] # tuning hyperparameter RF + oversampling
from sklearn.model_selection import RandomizedSearchCV

n_estimators = [int(x) for x in np.linspace(1, 200, 50)]
criterion = ['gini', 'entropy']
max_depth = [int(x) for x in np.linspace(2, 100, 50)]
min_samples_split = [int(x) for x in np.linspace(2, 20, 10)]
min_samples_leaf = [int(x) for x in np.linspace(2, 20, 10)]
hyperparameters = dict(n_estimators=n_estimators, criterion=criterion, max_depth=max_depth,
                        min_samples_split=min_samples_split, min_samples_leaf=min_samples_leaf)

rf = RandomForestClassifier(random_state=42)
rs = RandomizedSearchCV(rf, hyperparameters, scoring='roc_auc', random_state=1, cv=5)
rs.fit(X_train_over, Y_train_over)
eval_classification(rs)
```

```
Accuracy (Test Set): 0.90
Precision (Test Set): 0.50
Recall (Test Set): 0.52
F1-Score (Test Set): 0.51
roc_auc (test-proba): 0.90
roc_auc (train-proba): 0.98
roc_auc (crossval train): 0.9333301217501548
roc_auc (crossval test): 0.7095961997052507
```

# Modelling

## Ada Boost

```
[ ] from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
import numpy as np

# List of hyperparameter
hyperparameters = dict(n_estimators = [int(x) for x in np.linspace(start = 50, stop = 2000, num = 2000)], # Jumlah iterasi
                        learning_rate = [float(x) for x in np.linspace(start = 0.001, stop = 0.1, num = 200)],
                        algorithm = ['SAMME', 'SAMME.R']
                        )

# Init model
ab = AdaBoostClassifier(random_state=42)
ab_tuned = RandomizedSearchCV(ab, hyperparameters, random_state=42, cv=5, scoring='recall')
ab_tuned.fit(X_train,Y_train)

# Predict & Evaluation
eval_classification(ab_tuned)

Accuracy (Test Set): 0.90
Precision (Test Set): 0.56
Recall (Test Set): 0.22
F1-Score (Test Set): 0.32
roc_auc (test-proba): 0.89
roc_auc (train-proba): 0.91
roc_auc (crossval train): 0.9172245957881311
```

# Modelling

## XG Boost

```
from sklearn.model_selection import RandomizedSearchCV, GridSearchCV
[ ] import numpy as np

#Menjadikan ke dalam bentuk dictionary
hyperparameters = {
    'max_depth' : [int(x) for x in np.linspace(10, 110, num = 11)],
    'min_child_weight' : [int(x) for x in np.linspace(1, 20, num = 11)],
    'gamma' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'tree_method' : ['auto', 'exact', 'approx', 'hist'],

    'colsample_bytree' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'eta' : [float(x) for x in np.linspace(0, 1, num = 100)],

    'lambda' : [float(x) for x in np.linspace(0, 1, num = 11)],
    'alpha' : [float(x) for x in np.linspace(0, 1, num = 11)]
}

# Init
from xgboost import XGBClassifier
xg = XGBClassifier(random_state=42)
xg_tuned = RandomizedSearchCV(xg, hyperparameters, cv=5, random_state=42, scoring='recall')
xg_tuned.fit(X_train,Y_train)

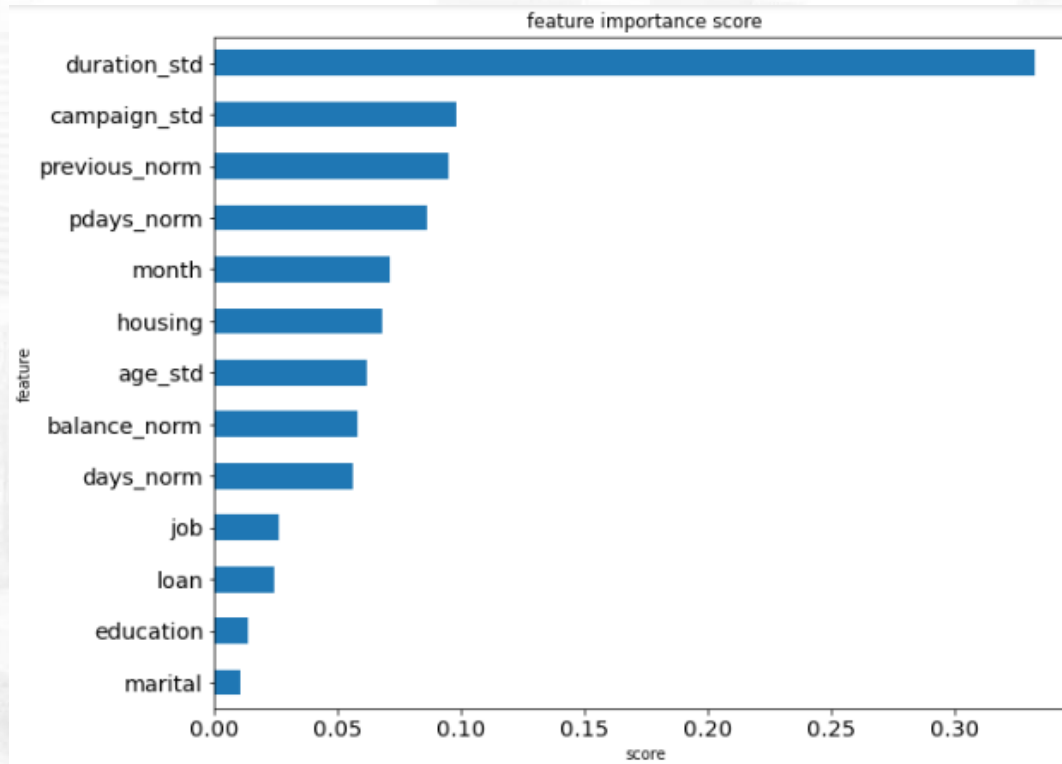
# Predict & Evaluation
eval_classification(xg_tuned)
```

```
Accuracy (Test Set): 0.91
Precision (Test Set): 0.59
Recall (Test Set): 0.37
F1-Score (Test Set): 0.45
roc_auc (test-proba): 0.92
roc_auc (train-proba): 0.99
roc_auc (crossval train): 0.9765877507321999
roc_auc (crossval test): 0.5891293229797601
```

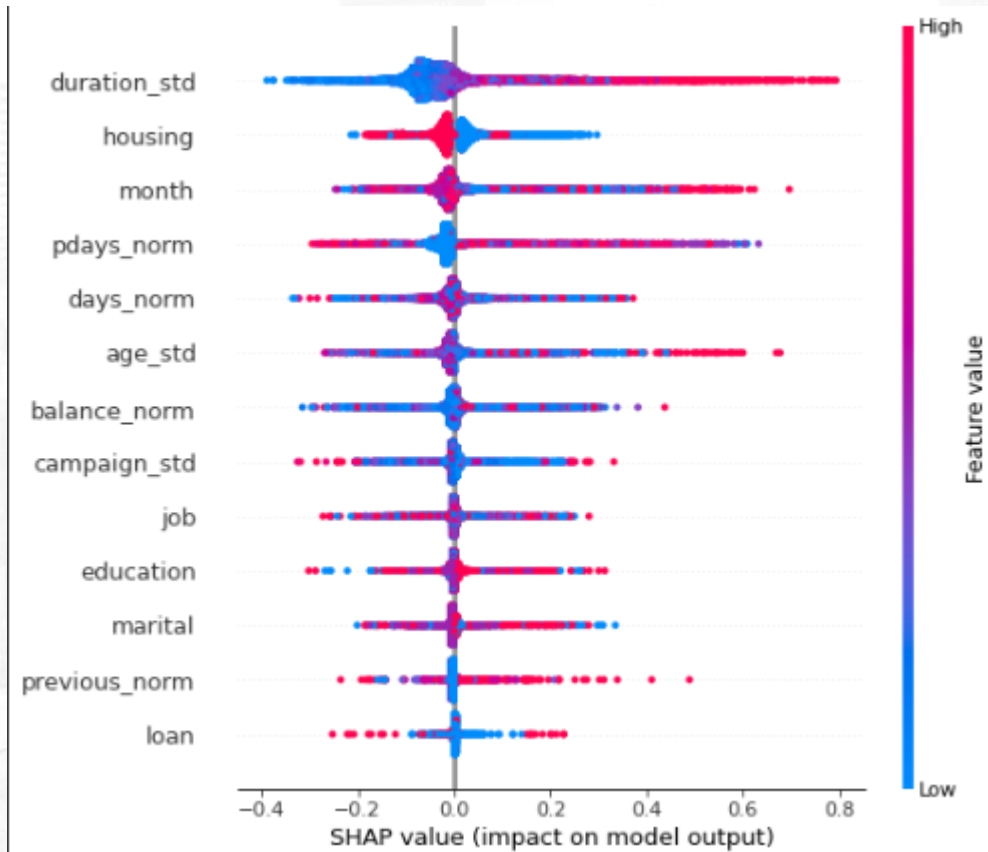
Ketika melakukan uji coba data dengan melakukan pemodelan, dapat disimpulkan bahwa diperoleh nilai Accuracy yaitu 0.90 dan gap antara test dan train kecil yaitu 0.02 pada saat menggunakan modeling Adaboost.



# Feature Importance



# Shap Value





# Business Recommendation

Berdasarkan Business Insight, berikut rekomendasi yang dapat diberikan:

- Bank perlu memberikan pembiayaan lebih terhadap biaya telpon agar dapat menyesuaikan dengan waktu yang telah ditetapkan berdasarkan business insight yaitu 2-9 menit.
- Membuat campaign produk deposito secara masif, baik melalui telpon maupun melalui website & media perbankan.