**PREDIKSI FINANCIAL DISTRESS DENGAN TEKNIK BAGGING, BOOSTING, DAN STACKING**

**LAPORAN UAS TEKNIS**

**KECERDASAN BUATAN**



ERWIN JELLY BARUS TOBING 10118702

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS KOMPUTER INDONESIA

2021

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc65731721)

[DAFTAR TABEL 4](#_Toc65731722)

[DAFTAR GAMBAR 4](#_Toc65731723)

[DAFTAR SCRIPT 4](#_Toc65731724)

[DAFTAR GRAFIK 5](#_Toc65731725)

[KATA PENGANTAR 6](#_Toc65731726)

[BAB 1 PENDAHULUAN 7](#_Toc65731727)

[1.1 Latar Belakang 7](#_Toc65731728)

[BAB 2 TINJAUAN PUSTAKA 8](#_Toc65731729)

[2.1 Support Vector Machine (SVM) 8](#_Toc65731730)

[2.2 Logistic Regression 8](#_Toc65731731)

[2.3 Artificial Neural Network (ANN) 9](#_Toc65731732)

[2.4 Adaptive Boosting 10](#_Toc65731733)

[2.5 Voting Classifier 10](#_Toc65731734)

[2.6 Bagging 11](#_Toc65731735)

[2.7 Boosting 11](#_Toc65731736)

[2.8 Stacking 12](#_Toc65731737)

[2.9 Confusion Matrix 12](#_Toc65731738)

[BAB 3 METODE PENELITIAN 13](#_Toc65731739)

[3.1 Import Library 13](#_Toc65731741)

[3.2 Import Dataset dan Menampilkan Tabel Serta Grafiknya 14](#_Toc65731742)

[3.3 Korelasi Fitur 15](#_Toc65731743)

[3.4 Memilih Target dan Fitur 16](#_Toc65731744)

[3.5 Imbalance Handling 16](#_Toc65731745)

[3.6 Standarisasi 16](#_Toc65731746)

[3.7 Split Dataset 17](#_Toc65731747)

[3.8 Mendefinisikan Base Model 17](#_Toc65731748)

[3.9 Bagging 17](#_Toc65731749)

[3.10 Boosting 18](#_Toc65731750)

[3.11 Stacking 18](#_Toc65731751)

[3.12 K-Fold 19](#_Toc65731752)

[3.13 Confusion Matrix 19](#_Toc65731753)

[BAB 4 HASIL DAN PEMBAHASAN 21](#_Toc65731754)

[4.1 Perbandingan Tabel 22](#_Toc65731755)

[4.2 Perbandingan Diagram 23](#_Toc65731756)

[BAB 5 KESIMPULAN DAN SARAN 24](#_Toc65731757)

[5.1 Kesimpulan 24](#_Toc65731758)

[5.2 Saran Pengembangan 24](#_Toc65731759)

[DAFTAR PUSTAKA 25](#_Toc65731760)

## DAFTAR TABEL

[Table 1 Dataset Bank Umum Syariah pada tahun 2011-2016 14](#_Toc65731587)

[Table 2 Korelasi antar fitur 16](#_Toc65731588)

[Table 3 Perbandingan akurasi teknik bagging, stacking, dan boosting 22](#_Toc65731589)

## DAFTAR GAMBAR

[Gambar 1 Hasil Oversampling 16](#_Toc65731598)

[Gambar 2 Hasil akurasi dengan k-fold 19](#_Toc65731599)

[Gambar 3 Hasil confusion matrix dan report model bagging 20](#_Toc65731600)

[Gambar 4 Hasil confusion matrix dan report model boosting 20](#_Toc65731601)

## DAFTAR SCRIPT

[Script 1 Import Library 13](#_Toc65731608)

[Script 2 Pemilihan Label dan Fitur 16](#_Toc65731609)

[Script 3 Imbalance Dataset Handling dengan Oversampling 16](#_Toc65731610)

[Script 4 Penyamaan Skala 17](#_Toc65731611)

[Script 5 Split data training dan data testing 17](#_Toc65731612)

[Script 6 Mendefinisikan Base Model 17](#_Toc65731613)

[Script 7 Training dan prediksi model bagging 18](#_Toc65731614)

[Script 8 Training dan prediksi model boosting 18](#_Toc65731615)

[Script 9 Training dan prediksi model stacking 18](#_Toc65731616)

[Script 10 Menghitung akurasi dengan k-fold 19](#_Toc65731617)

[Script 11 Confusion matrix dan report model bagging 19](#_Toc65731618)

[Script 12 Confusion matrix dan report model boosting 20](#_Toc65731619)

[Script 13 Confusion matrix dan report model stacking 21](#_Toc65731620)

## DAFTAR GRAFIK

[Grafik 1 Dataset Perusahaan Customer Good pada tahun 2010-2016 15](#_Toc65731682)

[Grafik 2 Perbandingan akurasi teknik bagging, stacking, dan boosting 23](#_Toc65731683)

## KATA PENGANTAR

Pertama sekali saya ucapkan puji dan syukur kepada Tuhan Yang Maha Esa sehingga saya dapat menyelesaikan laporan ini dengan baik. Adapun penyusunan laporan ini ditujukan guna memenuhi tugas UAS Teknis mata kuliah Kecerdasan Buatan.

Pada kesempatan ini, saya ingin mengucapkan terimakasih yang sebesar-besarnya kepada dosen kami yakni pak Bambang Siswoyo, Ir,.M.Si yang telah membantu dengan memberikan materi dan arahan dalam pengerjaan laporan ini.

Karena keterbatasan pengetahunan maupun pengalaman, saya yakin masih banyak kekurangan dalam laporan ini. Oleh karena itu, saya menerima saran dan kritik yang membangun demi kesempurnaan laporan ini. Atas segala perhatiannya saya sampaikan banyak terimakasih.

Bandung, 14 Februari 2021

Penulis

## PENDAHULUAN

### Latar Belakang

Financial distress merupakan suatu risiko melekat yang dapat dialami oleh setiap perusahaan. Financial distress dapat didefinisikan sebagai kondisi di mana aktivitas perusahaan terhambat dikarenakan mengalami masalah kesulitan keuangan. Perusahaan yang terindikasi mengalami kebangkrutan ditandai dengan gejala kesulitan keuangan.

Salah satu indikasi bahwa perusahaan mengalami financial distress adalah adanya kewajiban finansial yang tidak dapat dipenuhi (Parulian, 2007). Terkait dengan tidak terpenuhinya kewajiban finansial, parameternya adalah kegagalan perusahaan dalam memenuhi persyaratan yang ada dalam suatu kontrak utang. Lebih lanjut, financial distress juga ditandai dengan pengahapusan dan/atau pengurangan pembayaran dividen

Dari beberapa indikasi dari perusahaan tersebut dapat dilakukan prediksi untuk mencegah kebangkrutan dengan menggunakan model machine learning. Salah satu model yang paling sering digunakan untuk prediksi financial distress dan sudah cukup terkenal yaitu dengan ensemble learning.

Ensemble learning merupakan metode algoritma dalam machine learning dimana algoritma ini sebagai pencarian solusi prediksi terbaik dibandingkan dengan algoritma yang lain karena metode ensemble ini menggunakan beberapa algoritma pembelajaran untuk pencapaian solusi prediksi yang lebih baik daripada algoritma yang bisa diperoleh dari salah satu pembelajaran algoritma kosituen saja.

Evaluasi prediksi dari ensemble biasanya memerlukan banyak komputasi daripada evaluasi prediksi model tunggal (single model), jadi ensemble ini memungkinkan untuk mengimbangi poor learning algorithms oleh performasi lebih dari komputasi itu.

Adapun ensemble learning memiliki 3 teknik yang sudah cukup terkenal yaitu Bagging, Boosting, dan Stacking. Oleh karena itu pada kesempatan ini dilakukan studi untuk membandingkan prediksi kebangkrutan perusahaan-perusahan Polandia dengan teknik Bagging, Boosting, dan Stacking.

## TINJAUAN PUSTAKA

Dalam penelitian ini terdapat beberapa model pembelajaran mesin yang akan digunakan untuk prediksi. Beberapa *base model* dan *ensemble model* serta beberapa teknik yang akan dipakai yaitu seperti berikut.

### Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan sebuah algoritma klasifikasi untuk data linear dan non-linear. SVM menggunakan mapping non-linear untuk mentransformasikan training data awal ke dimensi yang lebih tinggi. Support Vector Machine (SVM) merupakan salah satu metode dalam supervised learning yang biasanya digunakan untuk klasifikasi (seperti Support Vector Classification) dan regresi (Support Vector Regression). Dalam pemodelan klasifikasi, SVM memiliki konsep yang lebih matang dan lebih jelas secara matematis dibandingkan dengan teknik-teknik klasifikasi lainnya. SVM juga dapat mengatasi masalah klasifikasi dan regresi dengan linear maupun non linear.

SVM digunakan untuk mencari hyperplane terbaik dengan memaksimalkan jarak antar kelas. Hyperplane adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai line whereas, fungsi yang digunakan untuk klasifikasi antas kelas dalam 3-D disebut plane similarly, sedangan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi di sebut hyperplane.

### Logistic Regression

Regresi logistik merupakan salah satu jenis regresi yang menghubungkan antara satu atau beberapa variabel independen (variabel bebas) dengan variabel dependen yang berupa kategori; biasanya 0 dan 1. Regresi logistik (kadang disebut model logistik atau model logit), dalam statistika digunakan untuk prediksi probabilitas kejadian suatu peristiwa dengan mencocokkan data pada fungsi logit kurva logistik. Metode ini merupakan model linier umum yang digunakan untuk regresi binomial. Seperti analisis regresi pada umumnya, metode ini menggunakan beberapa variabel prediktor, baik numerik maupun kategori. Misalnya, probabilitas bahwa orang yang menderita serangan jantung pada waktu tertentu dapat diprediksi dari informasi usia, jenis kelamin, dan indeks massa tubuh. Regresi logistik juga digunakan secara luas pada bidang kedokteran dan ilmu sosial, maupun pemasaran seperti prediksi kecenderungan pelanggan untuk membeli suatu produk atau berhenti berlangganan.

### Artificial Neural Network (ANN)

Artificial Neural Network Artificial (ANN) atau Jaringan Syaraf Tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Elemen kunci dari teknik ini adalah struktur sistem pengolahan informasi yang bersifat unik dan beragam untuk tiap aplikasi. Neural Network terdiri dari sejumlah besar elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya dalah masalah klasifikasi ataupun prediksi.

Cara kerja Neural Network dapat dianalogikan sebagaiman halnya manusia belajar dengan mengunakan contoh atau yang disebut sebagai supervised learning. Sebuah Neural Network dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, dan kemudian disempurnakan melalui proses pembelajaran. Proses belajar yang terjadi dalam sistem biologis melibatkan penyesuaian koneksi sinaptik yang ada antara neuron, dalam halnya pada Neural Network penyesuaian koneksi sinaptik antar neuron dilakukan dengan menyesuaikan nilai bobot yang ada pada tiap konektivitas baik dari input, neuron maupun output.

Neural Network memproses informasi berdasarkan cara kerja otak manusia. Dalam hal ini Neural Network terdiri dari sejumlah besar elemen pemrosesan yang saling terhubung dan bekerja secara paralel untuk memecahkan suatu masalah tertentu. Di sisi lain, komputer konvensional menggunakan pendekatan kognitif untuk memecahkan masalah; dimana cara pemecahan masalah haruslah sudah diketahui sebelumnya untuk kemudian dibuat menjadi beberapa instruksi kecil yang terstruktur. Instruksi ini kemudian dikonversi menjadi program komputer dan kemudian ke dalam kode mesin yang dapat dijalankan oleh komputer.

Neural Network dan algoritma komputer konvensional tidaklah saling bersaing tetapi saling melengkapi. Beberapa tugas atau masalah lebih cocok diselesaikan dengan pendekatan algoritmik seperti halnya operasi aritmatika, di sisi lain ada tugas-tugas yang lebih cocok untuk jaringan saraf, misalnya prediksi pergerakan data time-series. Bahkan, sejumlah besar tugas lainnya memerlukan sistem yang menggunakan kombinasi dari keddua pendekatan tersebut, dimana biasanya komputer konvensional digunakan untuk mengawasi Neural Network agar dapat memberikan kinerja maksimum.

### Adaptive Boosting

Metode AdaBoost merupakan salah satu algoritma supervised pada data mining yang diterapakan secara luas untuk membuat model klasifikasi. AdaBoost sendiri pertama kali diperkenalkan oleh Yoav Freund dan Robert Schapire(1995). Walaupun pada awalnya algoritma ini diterapkan pada model regresi, seiring dengan perkembangan teknologi komputer yang cepat, metode ini juga dapat diterapkan pada model statistik lainnnya. Metode adaBoost merupakan salah satu teknik ensamble dengan menggunakan loss function fungsi exponential untuk memperbaiki tingkat akurasi dari prediksi yang dibuat. Pada makalah ini akan akan dijelaskan penerapan metode AdaBoostdalam masalah pengklasifikasian dengan tujuan untuk memperbaiki tingkat akurasi model yang dibentuk.

Adaboost merupakan ensemble learning yang sering digunakan pada algoritma boosting. Boosting bisa dikombinasikan dengan classifier algoritma yang lain untuk meningkatkan performa klasifikasi. Tentunya secara intuitif, penggabungan beberapa model akan membantu jika model tersebut berbeda satu sama lain. Adaboost dan variannya telah sukses diterapkan pada beberapa bidang (domain) karena dasar teorinya yang kuat, presdiksi yang akurat, dan kesederhanaan yang besar.

### Voting Classifier

Voting adalah cara sederhana yang digunakan untuk menggabungkan beberapa algoritma machine learning. Voting merupakan salah satu cara paling sederhana untuk menggabungkan prediksi dari beberapa algoritme pembelajaran mesin. Voting classifier bukanlah klasifikasi yang sesungguhnya, melainkan pembungkus untuk kumpulan pengklasifikasi berbeda yang dilatih dan dinilai secara paralel untuk mengeksploitasi keanehan yang berbeda dari setiap algoritma.

Kita dapat melatih kumpulan data menggunakan algoritma dan ansambel yang berbeda kemudian untuk memprediksi hasil akhir dari sebuah prediksi diambil oleh suara terbanyak berdasarkan dua strategi berbeda:

* Hard voting / Majority voting: Hard voting adalah kasus paling sederhana dari voting mayoritas. Dalam hal ini, kelas yang menerima jumlah suara terbanyak akan dipilih Nc (y t). Di sini kami memprediksi label kelas y ^ melalui pemungutan suara mayoritas dari setiap pengklasifikasi.
* Soft voting : Dalam kasus ini, vektor probabilitas untuk setiap kelas yang diprediksi (untuk semua pengklasifikasi) dijumlahkan & dirata-ratakan. Kelas pemenang adalah kelas yang sesuai dengan nilai tertinggi (hanya disarankan jika pengklasifikasi dikalibrasi dengan baik).

### Bagging

Bagging merupakan metode yang dapat memperbaiki hasil dari algoritma klasifikasi machine learning dengan menggabungkan klasifikasi prediksi dari beberapa model. Hal ini digunakan untuk mengatasi ketidakstabilan pada model yang kompleks dengan kumpulan data yang relatif kecil. Bagging adalah salah satu algoritma berbasis ensemble yang paling awal dan paling sederhana, namun efektif. Bagging paling cocok untuk masalah dengan dataset pelatihan yang relatif kecil. Bagging mempunyai variasi yang disebut Pasting Small Votes. cara ini dirancang untuk masalah dengan dataset pelatihan yang besar, mengikuti pendekatan yang serupa, tetapi membagi dataset besar menjadi segmen yang lebih kecil. Penggolong individu dilatih dengan segmen ini, yang disebut bites, sebelum menggabungkannya melalui cara voting mayoritas.

### Boosting

Boosting merupakan cara untuk menghasilkan beberapa model atau penggolongan untuk prediksi atau klasifikasi, dan juga menggabungkan prediksi dari berbagai model ke dalam prediksi tunggal. Bagging adalah pendekatan iteratif untuk menghasilkan pengklasifikasi yang kuat, yang mampu mencapai kesalahan training seminimal mungkin dari sekelompok pengklasifikasi yang lemah. yang masing-masing hampir tidak dapat melakukan lebih baik daripada tebakan acak.

Boosting tidak dibatasi dengan secara algoritme. prosedur Boosting cukup sederhana. Misalkan weak learners akan bekerja pada distribusi data apa pun yang diberikan, dan mengambil biner tugas klasifikasi sebagai contoh.[4] sebagian besar algoritma Boosting terdiri dari iteratif learning pengklasifikasi lemah sehubungan dengan distribusi dan menambahkannya ke penggolong kuat akhir. Ketika mereka ditambahkan, mereka biasanya ditimbang dengan beberapa cara yang biasanya terkait dengan ketepatan weak learners. Setelah weak learners ditambahkan, data akan ditulis ulang: contoh yang salah dikuatkan dan contoh yang diklasifikasi dengan benar menurunkan berat badan. Dengan demikian, weak learners nantinya lebih fokus pada contoh-contoh bahwa weak learners sebelumnya salah klasifikasi.

### Stacking

Stacking merupakan cara untuk mengkombinasi beberapa model, dengan konsep meta learner. dipakai setelah bagging dan boosting. tidak seperti bagging dan boosting, stacking memungkinkan mengkombinasikan model dari tipe yang berbeda. Ide dasarnya adalah untuk train learner tingkat pertama menggunakan kumpulan data training asli, dan kemudian menghasilkan kumpulan data baru untuk melatih learner tingkat kedua, di mana output dari learner tingkat pertama dianggap sebagai fitur masukan sementara yang asli label masih dianggap sebagai label data training baru. Pembelajar tingkat pertama sering dihasilkan dengan menerapkan algoritma learning yang berbeda.

### Confusion Matrix

Confusion matrix juga sering disebut error matrix. Pada dasarnya confusion matrix memberikan informasi perbandingan hasil klasifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya. Confusion matrix berbentuk tabel matriks yang menggambarkan kinerja model klasifikasi pada serangkaian data uji yang nilai sebenarnya diketahui.

Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada confusion matrix. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Agar lebih mudah memahaminya, dapat dilihat pada penjelasan seperti berikut.

* True Positive (TP) : merupakan data positif yang diprediksi benar.
* True Negative (TN) : merupakan data negatif yang diprediksi benar.
* False Postive (FP) - Type I Error : merupakan data negatif namun diprediksi sebagai data positif.
* False Negative (FN) - Type II Error : merupakan data positif namun diprediksi sebagai data negatif.

## BAB 3 METODE PENELITIAN



### Import Library

Hal pertama yang dilakukan dalam proses simulai yaitu memasukkan library yang akan digunakan ke jupyter notebook.

1. # dataframe and array
2. import pandas as pd
3. pd.plotting.register\_matplotlib\_converters()
4. import numpy as np
6. # modula visualisasi data histogram
7. import matplotlib.pyplot as plt
8. %matplotlib inline
9. import seaborn as sns
10. from matplotlib import colors
12. # preprocessing
13. from sklearn.model\_selection import train\_test\_split
14. from sklearn.preprocessing import MinMaxScaler
16. # metric and evaluation
17. from sklearn.metrics import classification\_report
18. from sklearn.metrics import confusion\_matrix
19. from sklearn.metrics import accuracy\_score
20. from sklearn.model\_selection import cross\_val\_score, StratifiedKFold
22. # base model
23. from sklearn.svm import SVC
24. from sklearn.linear\_model import LogisticRegression
25. from sklearn.ensemble import RandomForestClassifier
26. from keras.models import Sequential
27. from sklearn.ensemble import AdaBoostClassifier
29. # ensemble machine learning model
30. from sklearn.ensemble import BaggingClassifier
31. from sklearn.ensemble import VotingClassifier
32. from mlxtend.classifier import EnsembleVoteClassifier
33. from sklearn.ensemble import StackingClassifier
35. # imbalance dataset
36. from collections import Counter
37. from imblearn.over\_sampling import SMOTE
39. # warnings filter & ignore all future warning
40. from warnings import simplefilter
41. simplefilter(action='ignore', category=FutureWarning)

Script 1 Import Library

### Import Dataset dan Menampilkan Tabel Serta Grafiknya

Dataset yang dipakai dalam perbandingan model yaitu dataset bank umum syariah di Indonesia pada tahun 2010-2016. Dataset disajikan dalam bentuk file csv agar dapat diimport.

Script import dataset dan tampilkan tabel serta keterangannya :

1. # membaca dataset csv
2. dataset = pd.read\_csv('dataset1016-ai.csv')
4. # menampilkan tabel dataset
5. pd.options.display.max\_rows = None
6. dataset

Table 1 menunjukkan 10 data teratas dari dataset bank umum syariah pada tahun 2010-2016. Table terdiri dari 6 kolom dimana kolom pertama sampai kelima digunakan sebagai fitur prediksi dan kolom Class sebagai target prediksi.

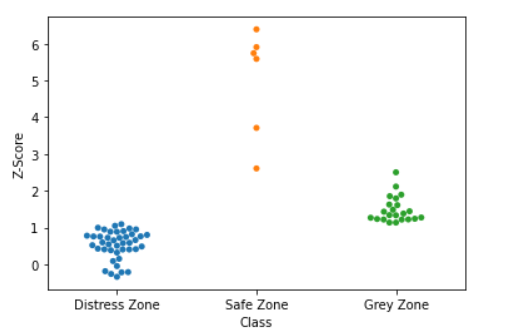
Table 1 Dataset Bank Umum Syariah pada tahun 2011-2016



Script tampilkan grafik dan keterangan :

1. # menampilkan grafik dataset
2. sns.swarmplot(x=dataset['Class'],
3. y=dataset['Z-Score'])

Grafik 1 menunjukkan sebaran kondisi financial distress bank umum syariah dimana sumbu Y menunjukkan nilai Z-Score dan sumbu X mewakil kondisi financial distress atau kelas dari dataset yaitu berupa Distress Zone, Safe Zone, dan Grey Zone. Dari sebaran plot dilihat bahwa sangat banyak perusahaan yang berada pada Distress Zone (biru), diikuti jumlah Grey Zone (hijau), dan jumlah bank pada kondisi Safe Zone (orange) relative sedikit,



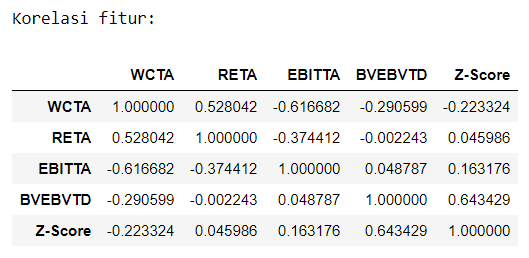
Grafik 1 Dataset Perusahaan Customer Good pada tahun 2010-2016

### Korelasi Fitur

Berikut script untuk menghitung dan menampilkan tabel korelasi antar fitur pada dataset.

1. dataset\_corr = dataset.corr()
2. print('\nKorelasi fitur:')
3. dataset\_corr

Table 2 Korelasi antar fitur



### Memilih Target dan Fitur

Setelah dataset diimportkan dilakukan pemilihan fitur atau variable prediksi dan juga targetnya.

1. y = dataset.Class
2. X = dataset.drop(['Class'], axis=1)

Script 2 Pemilihan Label dan Fitur

### Imbalance Handling

Fitur dan target yang telah dipilih kemudian disamakan sebaran kelas atau target (imbalanced dataset handling).

1. oversample = SMOTE()
2. X\_over, y\_over = oversample.fit\_resample(X, y)
4. print("Imbalance dataset  :", Counter(y))
5. print("Oversample dataset :", Counter(y\_over))

Script 3 Imbalance Dataset Handling dengan Oversampling



Gambar 1 Hasil Oversampling

### Standarisasi

Kemudian fitur dataset yang berupa *numeric countinue* disamakan skalanya dengan MinMaxScaler.

1. scaler = MinMaxScaler()
2. scaler.fit(X\_over)

Script 4 Penyamaan Skala

### Split Dataset

Selanjutnya dataset yang sudah diolah dibagi menjadi data training dan data testing dengan rasio data training 75% dan data testing 25%.

1. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_over, y\_over, test\_size=0.25, random\_state=0)

Script 5 Split data training dan data testing

### Mendefinisikan Base Model

Selanjutnya masuk pada tahapan membangun model yang terdiri data beberapa tahapan ekperimen dengan menggunakan algoritma ensemble learning yaitu :

1. RANDOM\_SEED = 0
3. svc\_clf = SVC()
4. lr\_clf = LogisticRegression()
5. ann\_clf = Sequential()
7. nama\_model = []
8. akurasi = []

Script 6 Mendefinisikan Base Model

### Bagging

Pembuatan model ensemble dengan menggunakan base model SVM dan Logistik Regression yang kemudian di voting dengan hard voting.

1. # membangun bagging model
2. bagging\_svc = BaggingClassifier(svc\_clf,  random\_state=RANDOM\_SEED)
3. bagging\_lr  = BaggingClassifier(lr\_clf,  random\_state=RANDOM\_SEED)
4. bagging\_ann = BaggingClassifier(ann\_clf,  random\_state=RANDOM\_SEED)
6. bagging\_models = []
7. bagging\_models.append(('BaggingSVC', bagging\_svc))
8. bagging\_models.append(('BaggingLogisticRegression', bagging\_lr))
10. # voting
11. bagging\_ensemble = VotingClassifier(bagging\_models)
12. # train
13. bagging\_ensemble.fit(X\_train, y\_train)
15. # prediksi
16. y\_pred1 = bagging\_ensemble.predict(X\_test)

Script 7 Training dan prediksi model bagging

### Boosting

Pembuatan model ensemble dengan menggunakan AdaBoost dan base estimator Logistic Regressor.

1. # training
2. boosting\_ensemble = AdaBoostClassifier(LogisticRegression(),n\_estimators=100, random\_state=0)
3. boosting\_ensemble.fit(X\_train, y\_train)
5. # prediksi
6. y\_pred2 = boosting\_ensemble.predict(X\_test)

Script 8 Training dan prediksi model boosting

### Stacking

Base model Logistic Regression dan SVM distacking dengan menggunakan Logistic Regression sebagain final estimator dan SVM sebagai base estimators.

1. Base\_models = []
2. Base\_models.append(('LogisticRegression', lr\_clf))
3. Base\_models.append(('SVM', svc\_clf))
5. # meta classifier
6. lr = LogisticRegression(random\_state=RANDOM\_SEED)
8. # membangun stacking
9. stacking\_ensemble = StackingClassifier(estimators=Base\_models, final\_estimator=lr)
11. # training model
12. stacking\_ensemble.fit(X\_train, y\_train)
14. # prediksi target
15. y\_pred3 = stacking\_ensemble.predict(X\_test)

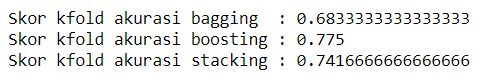
Script 9 Training dan prediksi model stacking

### K-Fold

Selanjutnya ketiga model ensemble yang telah dibangun diukur akurasinya dengan k-fold validasi.

1. kf = StratifiedKFold(n\_splits=10, shuffle=True)
3. score1 = cross\_val\_score(bagging\_ensemble, X\_test, y\_test, cv=kf, scoring='accuracy').mean()
4. score2 = cross\_val\_score(boosting\_ensemble, X\_test, y\_test, cv=kf, scoring='accuracy').mean()
5. score3 = cross\_val\_score(stacking\_ensemble, X\_test, y\_test, cv=kf, scoring='accuracy').mean()
7. print("Skor kfold akurasi bagging  : {}".format(score1))
8. print("Skor kfold akurasi boosting : {}".format(score2))
9. print("Skor kfold akurasi stacking : {}".format(score3))

Script 10 Menghitung akurasi dengan k-fold



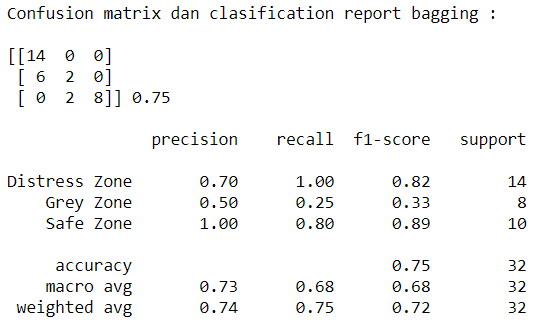
Gambar 2 Hasil akurasi dengan k-fold

### Confusion Matrix

Script 11 mencari dan menampilkan confusion matrix dari model bagging.

1. conf\_matrix = confusion\_matrix(y\_test, y\_pred1)
2. accuracy1 = accuracy\_score(y\_test, y\_pred1)
3. nama\_model.append("Bagging")
4. akurasi.append(accuracy1)
6. print("Confusion matrix dan clasification report bagging : \n")
7. print(conf\_matrix, accuracy1, '\n')
8. print(classification\_report(y\_test, y\_pred1))

Script 11 Confusion matrix dan report model bagging

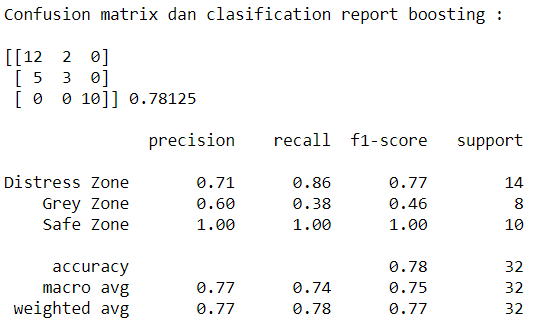


Gambar 3 Hasil confusion matrix dan report model bagging

Script 12 mencari dan menampilkan confusion matrix dari model bagging.

1. conf\_matrix = confusion\_matrix (y\_test, y\_pred2)
2. accuracy2 = accuracy\_score (y\_test, y\_pred2)
3. nama\_model.append ( "Boosting" )
4. akurasi.append (accuracy2)
6. print ( "Confusion matrix dan classification report boosting: \ n" )
7. print (conf\_matrix, accuracy2,  '\ n' )
8. print (classification\_report (y\_test, y\_pred2))

Script 12 Confusion matrix dan report model boosting

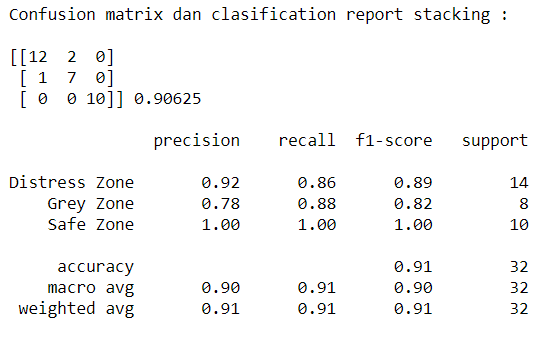


Gambar 4 Hasil confusion matrix dan report model boosting

Script 13 mencari dan menampilkan confusion matrix dari model stacking.

1. conf\_matrix = confusion\_matrix (y\_test, y\_pred3)
2. accuracy3 = accuracy\_score (y\_test, y\_pred3)
3. nama\_model.append ( "Stacking" )
4. akurasi.append (accuracy3)
6. print ( "Confusion matrix dan classification report stacking: \ n" )
7. print (conf\_matrix, accuracy3,  '\ n' )
8. print (classification\_report (y\_test, y\_pred3))

Script 13 Confusion matrix dan report model stacking



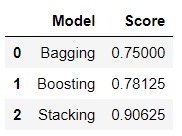
Gambar 5 Hasil confusion matrix dan report model stacking

## HASIL DAN PEMBAHASAN

Dari hasil evaluasi model didapatkan data akurasi dari setiap model yang dibangun terhadap dataset yang digunakan. Hasil dari perhitungan akurasi model dengan teknik bagging, stacking, dan boosting pada dataset Bank Umum Syariah dapat dilihat dengan tabel dan diagram seperti berikut.

### Perbandingan Tabel

Table 3 Perbandingan akurasi teknik bagging, stacking, dan boosting



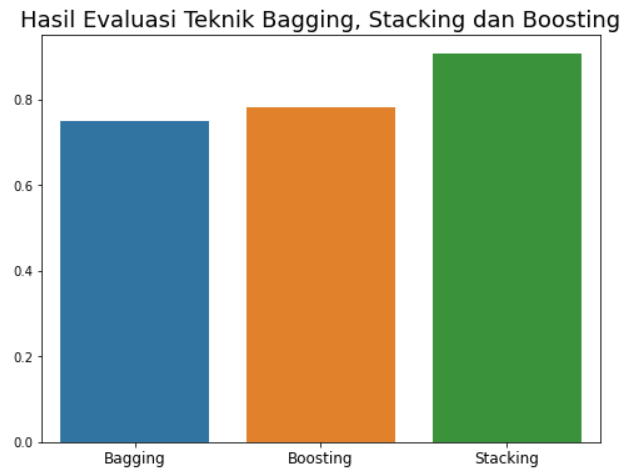
Keteranganan dan script :

1. # tabel hasil evaluasi model berdasarkan akurasi
2. data = {'Model':nama\_model, 'Score':akurasi}
3. df\_perbadingan = pd.DataFrame(data)
4. df\_perbadingan

Pada Table 3 menunjukkan perbandingan akurasi ensemble dengan teknik bagging, teknik boosting, dan teknik stacking pada dataset Bank Umum Syarian.

* Untuk teknik bagging pada memiliki akurasi 75%
* Untuk teknik boosting pada memiliki akurasi 78%
* Untuk teknik stacking pada memiliki akurasi 90%

### Perbandingan Diagram



Grafik 2 Perbandingan akurasi teknik bagging, stacking, dan boosting

Keteranganan dan script :

1. # grafik batang perbandingan hasil akurasi model
2. plt.figure(figsize=(8,6))
3. plt.rcParams['axes.labelsize'] = 14
4. plt.rcParams['axes.titlesize'] = 18
5. plt.rcParams['xtick.labelsize'] = 12
7. plt.title("Hasil Evaluasi Teknik Bagging, Stacking dan Boosting")
9. sns.barplot(x=nama\_model, y=akurasi)
10. plt.show()

Pada Grafik 3 menunjukkan perbandingan akurasi ensemble dengan teknik bagging, teknik boosting, dan teknik stacking pada dataset Bank Umum Syarian.

## KESIMPULAN DAN SARAN

### Kesimpulan

Beberapa poin kesimpulan yang dapat ditarik dari penelitian ini seperti berikut.

1. Model ensemble dibanguna dengan menggunakan base model Logistic Regression, ANN, dan SVM dengan teknik bagging, boosting, dan stacking.
2. Model diuji pada tiga dataset Bank Umum Syariah pada tahun 2010-2016.
3. Hasil akhir dari simulasi model yang dilakukan didapat hasil rata-rata performa model stacking lebih baik dibanding boosting dan bagging, serta model boosting sedikit lebih baik dibanding model bagging.

### Saran Pengembangan

Untuk saran yang terpikir oleh saya yaitu lebih berfokus pada pembanguan servernya. Menurut saya server secara Cloud Computing sangat cocok untuk pembangunan teknologi interopabilitas dimana, Cloud Computing sangat mendukung pertukaran informasi dengan lebih mudah dan cepat. Selain itu pembangunan server virtual juga lebih menghemat biaya dibanding apabila hendak membangun server fisik.

## DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | [Online]. Available: http://eprints.umg.ac.id/296/2/BAB-1.pdf. [Accessed July 2020]. |
| [2] | [Online]. Available: https://staf.ulm.ac.id/rezafaisal/2016/10/13/ensemble-learning-bagging/. [Accessed July 2020]. |
| [3] | [Online]. Available: https://id.wikipedia.org/wiki/Metode\_ensemble. [Accessed July 2020]. |
| [4] | [Online]. Available: https://medium.com/@arifinrio95/logistic-regression-concept-not-formula-derivation-or-algorithm-4295792975a1. [Accessed July 2020]. |
| [5] | [Online]. Available: https://id.wikipedia.org/wiki/Metode\_ensemble. [Accessed July 2020]. |
| [6] | [Online]. Available: https://medium.com/@saugata.paul1010/ensemble-learning-bagging-boosting-stacking-and-cascading-classifiers-in-machine-learning-9c66cb271674. [Accessed July 2020]. |
| [7] | [Online]. Available: https://www.kaggle.com/merishnasuwal/breast-cancer-prediction-dataset?select=Breast\_cancer\_data.csv. |
| [8] | [Online]. Available: https://www.kaggle.com/c/companies-bankruptcy-forecast/data?select=bankruptcy\_Train.csv. |