Klasifikasi email multi kelas menggunakan ensemble bagging

[1]Ali Helmut , [2]Adiwijaya , [3]Danang Triantoro Murdiansyah.

Fakultas Informatika, Universitas Telkom, Bandung

[1]alihelmut@student.telkomuniversity.ac.id,
[2]adiwijaya@telkomuniversity.ac.id,
[3] danangtri@telkomuniversity.ac.id

Abstrak

Email merupakan teknologi komunikasi yang umum dalam kehidupan modern ini. Semakin banyak email yang kita terima semakin sulit dan membutuhkan waktu untuk memilah. salah satu solusi untuk mengatasi masalah ini dengan cara membuat model matematis menggunakan pembelajaran mesin untuk memilah email berdasarkan konteks tertentu. Setiap jenis pembelajaran mesin dan distribusi data menghasilkan performansi yang berbeda. Ensemble merupakan suatu metode untuk megabungkan beberapa model menjadi satu kesatuan untuk mendapatkan performansi yang lebih baik. maka pada penelitian kami kami mencoba mengkombinasikan model pembelajaran, sampling dan beberapa kelas data untuk mendapatkan pengaruh bagging dan voting terhadap performansi macro average f1 score suatu model ensemble dan membandingkan dengan model *non-ensemble*. Hasil penelitian ini menunjukan sensitifitas Naïve Bayes terhadap data tak imbang terbantu oleh bagging dan voting dengan delta performansi 0.0001 – 0.0018, logistic regresi memiliki kenaikan performansi relative rendah untuk bagging dan voting dengan delta performansi 0.0001-0.00015, dan voting decision tree memiliki performansi yang terbebankan oleh Naïve Bayes dengan delta performansi -0.01.

Kata Kunci : Ensemble Learning, Bagging , Email-klasifikasi,

Abstract

Email is a common communication technology in modern life. The more emails we receive the more difficult and require time to sort. one solution to overcome this problem is by creating a mathematical model using machine learning to sort email based on certain contexts. Each type of machine learning and data distribution results in different performance. Ensemble is a method for combining several models into a single unit to get better performance. then in our study we tried to combine the learning model, sampling and some class data to get bagging and voting participation on the average macro performance of the f1 score of an ensemble model and compare it with the non-ensemble model. The results of this study indicate the sensitivity of Naïve Bayes to unbalanced data helped by bagging and voting with delta performance beetween 0.0001-0.0018, logistic regression increases the relatively low performance for bagging and voting with delta performance beetween 0.0001-0.00015, and the voting decision tree has the performance paid for by Naïve Bayes with delta performance -0.01.

Keywords — Ensemble learning, Bagging , Klasifikasi email

1. PENDAHULUAN

Teknologi komunikasi yang terus berkembang saat ini telah merubah cara masyarakat bertukar infomasi. Dimana saat ini kita dapat bertukar informasi dengan cepat, mudah dan murah. Salah satu teknologi komunikasi yang paling umum digunakan adalah email atau dalam Bahasa Indonesia disebut pesan elektronik. Dalam kehidupan modern ini, setidaknya setiap orang yang menggunakan internet memiliki satu buah akun email. Email merupakan suatu hal yang sangat penting untuk komunikasi dan berbagi informasi untuk sebagian besar masyarahat [1]. Dimana email biasa digunankan oleh kelompok atau organisasi untuk berbagi informasi dan sering juga digunakan untuk promosi sebuah produk. Untuk pengguna personal biasa digunakan untuk komunikasi formal seperti mengirimkan tugas kepada dosen, melaporakan hasil pekerjaan kepada atasan dan bisa digunakan sebagai identitas seseorang didalam dunia maya

Email terus mengalami peningkatan tiap tahunnya. Total email personal dan email bisnis yang dikirim tiap harinya mencapai 281 milyar pada tahun 2018, dan diperkirakan pada akhir tahun 2022 menyentuh angka 333 milyar [2]. Seperti yang telah dijelaskan diatas, email sering digunakan sebagai identitas di dunia maya. Setiap kali

kita mendaftarkan diri untuk menggunakan suatu fasilitas di internet seperti aplikasi social, aplikasi hiburan dan aplikasi lainnya sering kali kita diharuskan mengisi identitas diri dan juga email kita. Para penyedia fasilitas tersebut sering kali menawarkan kita untuk berlangganan informasi terbaru yang akan dikiriman via email, dan sering kali kita tak sadar menyetujuinya. Semakin sering kita menggunakan email baik untuk komunikasi personal maupun untuk syarat penggunakan suatu aplikasi, maka semakin banyak email yang akan kita terima setiap harinya

Beberapa penyedia email client memberikan akses kepada pengguna untuk memindahkan email ke folder tertentu secara manual. Namun Semakin banyak email yang kita terima setiap harinya, maka semakin sulit untuk memilah antara email penting dengan email yang kurang penting dan juga semakin banyak waktu yang kita butuhkan. Sekitar 46% pegawai yang menerima lebih dari serratus email perharinya menghabiskan satu jam bahkan lebih untuk mencari email penting dari kumpulan email yang tidak terorganisir [3].

Memindahkan email secara manual bukanlah jalan keluar dari masalah ini, maka kita membutuhkan suatu mesin yang dapat memilah email dengan sendirinya menjadi beberapa kategori. Namun email email tersebut harus dipisah berdasarkan apa? Beberapa penyedia email client memisahkan email dengan kategori kategori tertentu. Ada yang memisahkan email dengan melihat siapa pengirimnya, seperti teman, keluarga, rekan kerja dan sebagainya, ada juga yang memisahkan email berdasarkan konteks nya seperti yang dilakukan oleh aplikasi Google Mail. Dimana email dipisah menjadi 6 kategori yaitu kategori personal, kategori sosial , kategori update, kategori promosi, kategori forum dan juga spam.

Dengan cara memisahkan email menjadi beberapa kategori seperti yang dilakukan oleh beberapa email client diatas memudahkan pengguna untuk mencari email email yang mereka inginkan. Namun bagaimana caranya membuat suatu mesin yang memiliki kemampuan tersebut ? salah satu caranya yaitu dengan menggunakan metode pembelajaran mesin. Pembelajaran mesin merupakan suatu metode matematis yang dapat membuat suatu mesin memiliki kecerdasan tanpa diprogram secara eksplisit oleh programmer. Secara umum pembelajaran mesin dibedakan menjadi tiga, yaitu pembelajaran supervised learning atau pembelajaran terpimpin, unsupervised learning atau pembelajaran tidak terpimpin dan reinforcement learning atau pembelajaran penguatan.

Dengan menggunakan metode pembelajaran mesin terarah kita cukup memberikan data yang telah diberikan label/kelas (data latih) kepada program tersebut, dan program akan mempelajari pola dari data terhadap label dari seluruh data yang kita berikan. Proses ini biasa disebut proses belajar. Setelah proses pembelajaran selesai kita dapat memberikan data baru tak berlabel kepada mesin tersebut dan mesin tersebut akan mengklasifikasikan data baru tersebut kepada label/kelas berdasarkan pola yang telah dipelajari dari data latih yang kita berikan.

Metode pembelajaran mesin terarah secara umum menghasilkan kecerdasan yang baik jika distribusi data latih setiap labelnya berjumlah sama. Namun hal tersebut dalam dunia nyata sangatlah langka, sepertihalnya email yang kita terima setiap harinya. jumlah email promosi dan social yang kita terima sering kali lebih banyak dibandingkan email personal yang kita terima.

Beberapa metode untuk menyelesaikan masalah ini yaitu metode oversampling dan undersampling yang akan dijelaskan lebih dalam dibab selanjutnya. Setiap metode sampling memiliki kelebihan dan kekurangannya masing masing begitu juga dengan metode pembelajaran mesin. Metode manakah yang lebih baik? Setiap metode menghasilkan perfomansi yang lebih baik dibanding metode lainnya untuk kasus tertentu dan juga berlaku sebaliknya. Jika setiap metode baik mengapa kita tidak menggunakan semuanya? Dan jika kita akan mengkombinasikannya, bagaimana caranya ?

Kita dapat mengkombinasikan beberapa model pembelajaran mesin yang berbeda menjadi satu kesatuan dengan model dengan menggunakan Teknik ensemble/ansamble. Dimana setiap model melakukan proses pembelajaran secara terpisah, dan setelah proses pembelajaran terpisah selesai dilakukan proses aggregation/penyatuan. diamana teknik ini biasa disebut Teknik ensemble bagging (boostrap aggregating).

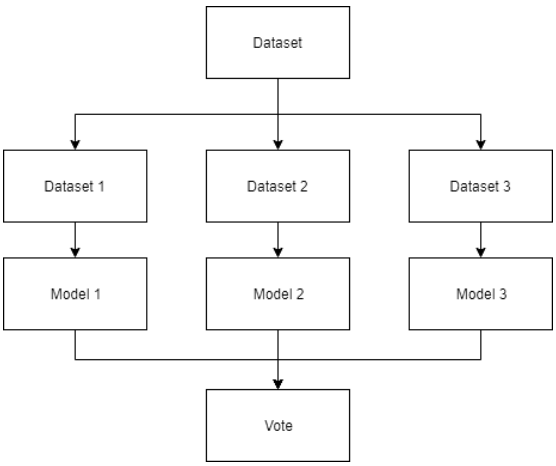
Secara umum ensemble meningkatkan performansi, namun apakah kenaikan performansi tersebut bersifat pasti? Maka pada penelitian ini kami mencoba membuat beberapa model dengan mengkombinasikan beberapa Teknik sampling, dan beberapa estimator dan melihat delta performansi antara ensemble dengan non-ensemble dengan matrik performansi *macro average f1 score.*

2. STUDI TERKAIT

2.1 Ensemble Learning

Ensemble Learning merupakan sebuah teknik untuk mengkombinasikan beberapa model yang sudah terlatih untuk menyelesaikan suatu permasalahan. Dimana teknik ini bertujuan untuk meningkatkan perfomansi dan menghindari overfiting [4].

Salah satu metode ensemble yang umum digunakan adalah metode ensemble bagging (boostrap aggregating). Pada metode bagging ini terdapat dua proses utama yaitu boostrap dan aggregating [7] dapat dilihat dari Gambar 1. Proses pertama yaitu bootsrap, dimana pada tahap ini dilakukan proses pembelajaran beberapa base model secara terpisah dengan data atau metode yang berbeda pada setiap model nya, Sehingga setiap model memiliki kecerdasan yang berbeda satu sama lain. Selanjutnya merupakan proses aggregating atau penyatuan. Pada tahap ini suatu finisher model mempelajari output dari setiap base model terhadap suatu data latih yang sama.



Gambar 1. Ensemble Bagging

2.2 Estimator.

Secara umum pembelajaran mesin dapat dibedakang menjadi tiga jenis jika dilihat dari bentuk atau proses belajar, yaitu model statistik, geometri, dan logika. Sehingga pada penelitian ini kami memilih 3 jenis estimator dimana setiap estimator mewakili ketiga bentuk pembelejaran mesin tersebut, diamana untuk Naïve Bayes[8] mewaikili statistika karena menggunakan perhitungan probabilitas, Random Forest[9] mewakili logika karena bentuk akhir merupakan fungsi logika percabangan dan Logistik regresi[10] mewakili geometri karena *hyperplane* sebagai batas antar kelas.

2.3 Sampling

- Under sampling

  Undersampling merupakan salah satu teknik yang berkerja dengan cara menghilangkan beberapa data kelas mayoritas , sehingga jumlah kelas mayoritas tersebut seimbang dengan data kelas minoritas [6]. Pada penelitian ini kami menggunakan random under sampling

- Over sampling

  Mengurangi data kelas mayoritas untuk menyeimbangkan jumlah data antar dapat menghilangkan banyak informasi dan membuat model mengalami *underfitting*. Maka salah satu solusi yang ditawarkan untuk menangai kekurangan teknik *undersampling* dengan cara melakukan proses yang berlawanan terhadap teknik tersebut yang biasa disebut teknik *Oversampling.* O*versampling* merupakan suatu teknik untuk menyeimbangkan jumlah data antar kelas dengan menduplikasi beberapa data kelas minoritas [6]. Pada penelitian ini kami menggunakan Teknik oversampling kmeansmote [11].

2.4 Pra proses

Fitur dalam bentuk teks dibersihkan dengan cara menghilangkan non-alfabet, transformasi teks menjadi non-kapital, *stemming* dan menghilangkan *stopwords.* Data dalam bentuk text tidak dapat diproses secara langsung oleh mesin, sehingga data di transformasi menjadi bentuk *bag of word* menggunakan metode TFIDF.

2.5 Normalisasi dan standarisasi

Metode pembelajaran mesin akan lebih mudah memperlajari pola data saat distribusi data dalam bentuk normal, maka dibutuhkan proses normalisasi data latih. data distandarisasi dengan menggunakan rumus (2) dan dinormalisasi menggunakan mean max (1)

X_new = (X - X_min) / (X_max – X_min (1)
Z_i = (X_i – X_mean) / standar deviasi (2)

2.6 Fitur seleksi

Untuk melihat pengaruh suatu fitur terhadap suatu label dapat dilakukan proses perhitungan derajat kebebasan. Pada penelitian ini kami
menggunakan metode chi (4) squre untuk melihat derajat kebebasan setiap fitur dan memilih 250 fitur dengan nilai tertinggi.

$$X^2 = \sum(O - E)^2 \qquad (3)$$

Dimana O merupakan Frekuensi obserfasi dan E merupakan ekspetasi frekuensi

2.7 Performansi

Pada penelitian kami, matrik performansi model dilihat dari macro average f1 score. Dan untuk melihat pengaruh performansi ensemble dilakukan perhitungan delta performansi (4). Dalam proses perhitungan performansi data latih, digunakan metode cross validasi sehingga performansi data latih didapat dari rata rata macro average f1 score setiap segmen.

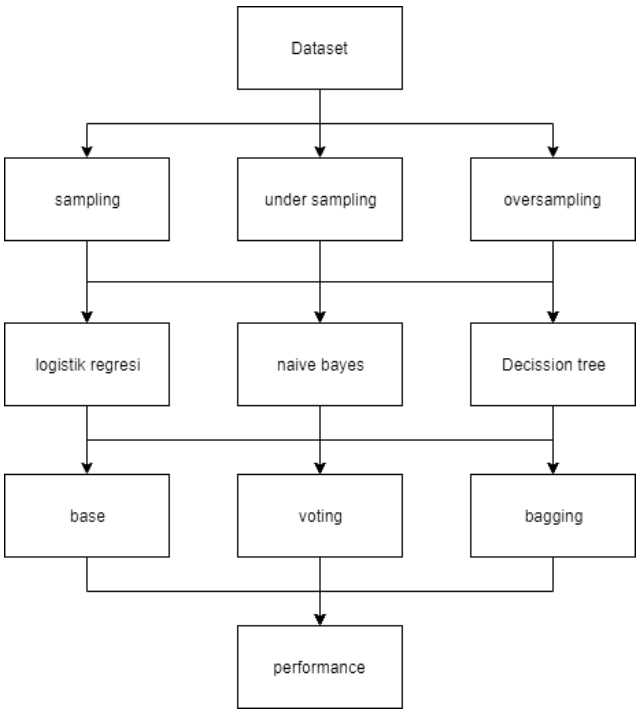
$$\Delta P = P_e - P_i \qquad (4)$$

Dimana ΔP merupakan delta performansi, P_e merupakan performansi ensemble dan P_i merupakan performansi non ensemble

3. SISTEM YANG DIBAGUN

3.1 Gambaran umum penelitian

Penelitian ini dilakukan dengan arsitektur pada Gambar 2. Data latih di transofrmasikan menjadi 3 data yaitu data *non-sampling* , data *under sampling*, dan data *over sampling.* Setiap data latih tersebut dilakukan kombinasi terhadap 3 estimator sehingga menghasilkan 9 model. Setiap model dilakukan perhitungan performansi, yaitu performansi non-ensemble, performansi ensemble voting dan performansi ensemble bagging. Dan proses terakhir yaitu proses perhitungan delta performansi (4)



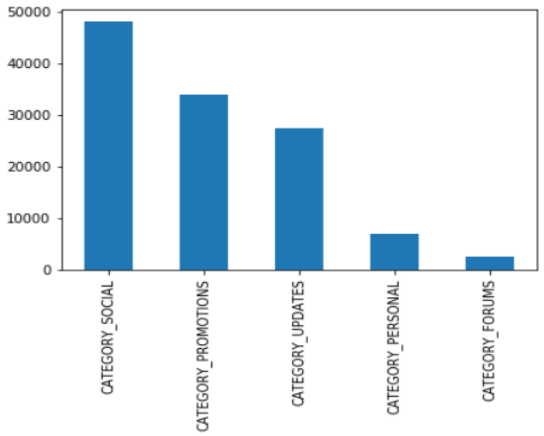
Gambar 2 Model yang dibangun

3.2 Dataset

Dataset pada penelitian ini didapat dari salah satu perusahaan email di kota Bandung Indonesia yaitu PT Proofn Indonesia, dimana data tersebut didapat dari Gmail para pekerjanya.

Dataset terdiri dan 5 kelas, yaitu kelas personal, update, promosi, sosial dan forum. Kelas personal merupakan email antara personal dengan personal. Kelas update merupakan email personal *auto generated update* termasuk konfirmasi, tahigan, resep dan peryataan. Kelas social merupakan email dari media social. Kelas forum merupakan email dari grup daring. Dan kelas promosi merupakan email promosi, diskon dan email marketing lainnya.

Data email tersebut terdiri dari beberapa informasi yang didapat dari header email, seperti kelas, waktu terkirim, judul email, potongan isi email, berhenti berlangganan, dan tipe isi email. Dan berikut distribusi kelas data yang didapat pada Gambar 3.



Gambar 3. Distribusi kelas

3.3 Pembagian Kelas

Pada penelitian ini, kelas di kombinasikan menjadi 4 uji kasus. Kasus pertama menggunakan kelas asli (_label), yaitu kelas personal, forum, social, update , promosi. Kasus kedua menggunakan dua kelas (_label_a), yaitu kelas personal dan lainnya. Kasus ketiga menggunakan du akelas (_label_c), yaitu kelas personal dan update sebagai sebagai kelas pertama, dan kelas kedua lainnya. Dan uji kasus terakhir tiga kelas (_kelas_c), yaitu kelas personal dan update sebagai kelas pertama, kelas social dan forum sebagai kelas kedua, dan promosi sebagai kelas ketiga.

3.4 Pra proses data

Data terdiri dari beberapa variable kategorikal dan satu variabel text, dimana variable data yang digunakan sebagai berikut:

Label	: kelas data
Date	: Waktu email dikirim
Sender	: email pengirim
Subject	: judul email
Snippet	: potongan isi email (tidak digunakan/)
Unsubscribe	: Boolean header email terdapat unsubscribe atau tidak
Mime	: mime type email (text, pdf, rar dan lainnya)

3.4.1. Variable Text:

Variable *subject* merupakan variable text, sehingga dibutuhkan proses sebelum diolah oleh estimator, maka berikut pra proses yang dilakukan untuk memtransformasikan text menjadi bentuk vector:
Menghilangkan non alfabetik, mentransformasikan text menjadi non kapital, mendeteksi Bahasa text, menghilangkan *stopwords* dan melakukan *stemming*. Untuk Bahasa non inggris dan non Indonesia proses penghilangan *stopwords* dan *stemming* tetap menggunakan data Bahasa inggris. Kemudian data transformasikan menggunakan TFIDF.

3.4.2 Variable Non Text:

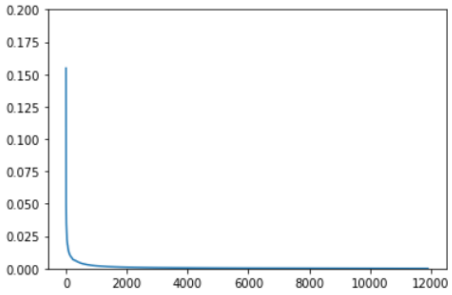
Berikut ini variable non text dan informasi yang diambil dari variable tersebut:

Date : Hari dan Jam
Sender : domail dan Boolean email mengandung string "noreply"
Mime : menghilangkan informasi yang duplikat

Setelah Proses diatas dilakukan, selanjutnya dilakukan proses *Scalling* data tiap variable dengan rentang 0 sampai 1 dengan cara membagi setiap variable dengan nilai maksimal, proses tersebut dilakukan terhadap setiap variable kecuali variable text. Dan selanjutnya setiap variable dilakukan proses One Hot encoder untuk menghilangkan bias kategorikal.

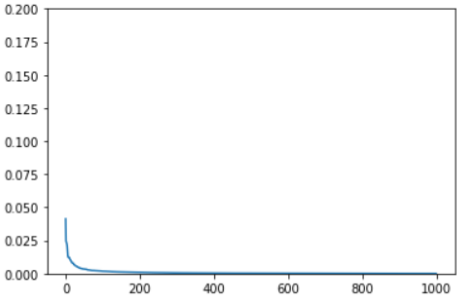
3.5 Fitur seleksi

Data telah ditransformasikan menjadi bentuk matrix (bag of words + kategorikal) memiliki jumlah dimensi yang sangat besar, maka selanjutnya dilakukan proses seleksi fitur dengan menggunakan perhitungan Chi Square terhadap kelas uji 1 (_label). Dari proses perhitungan *chi square* tersebut didapat informasi pada gambar dua dimana terdapat 12.000 fitur ada sekitar 2000 fitur dengan nilai derajat kebebasan diatas 0 . Maka fitur diseleksi dengan memilih 1000 fitur dengan derajat tertinggi.

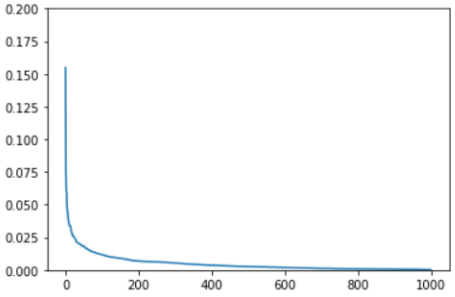


Gambar 4 Chi Square terhadap _label

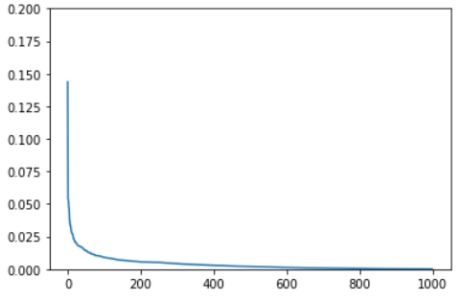
Setelah data direduksi menjadi 1000 fitur, dilakukan proses yang sama kembali. proses di atas hanya dilakukan perhitungan data awal terhadap kelas uji 1 (_label), dan proses selanjut nya dilakukan proses perhitungan terhadap keempat kelas uji (_kelas,_kelas_a,_kelas_b,_kelas_c) sehingga menghasilkan grafik pada Gambar 5-8. Dari keempat informasi yang didapat dari grafik pada Gambar 5-8 ditarik kesimpulan terdapat untuk mereduksi data dari 1000 fitur menjadi 250 fitur.



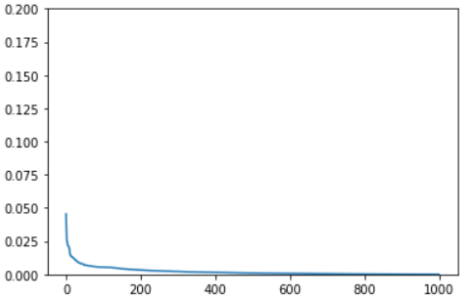
Gambar 5. Chi Square terhadap _Label



Gambar 6. Chi Square terhadap _label_a



Gambar 7. Chi Square terhadap _Label_b



Gambar 8. Chi Square terhadap _Label_c

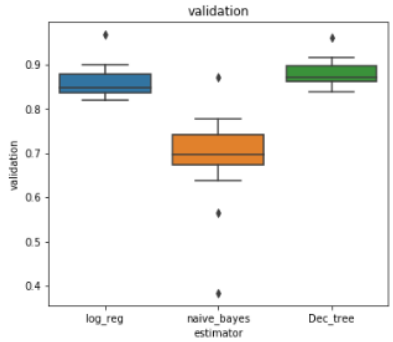
3.6 Sampling dan Pembagian data

Data latih dilakakukan over sampling dengan metode K-Means-Smote dan undersampling menggunakan pemilihan secara acak. Kemudian setiap data dibagi menjadi dua bagian, yaitu data latih dan data uji dengan perbandingan 80:20, dan selanjutnya data latih dibagi kembali dengan perbandingan 80:20 sebagai data latih dan data validasi menggunakan metode Cross Validation dengan jumlah 12 sektor.

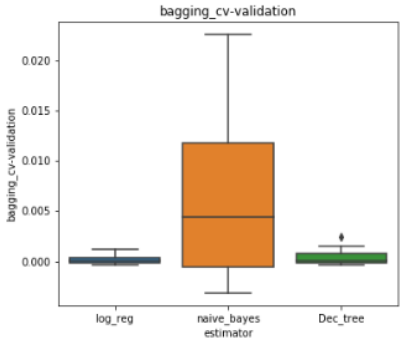
3.7 Estimator dan Model

Penelitian ini menggunakan 3 jenis estimator, yaitu Logistic Regresi, Decision Tree dan Naïve Bayes. Semua estimator menggunakan hyper parameter *default Sklaern 0.21.3.* model merupakan kombinasikan sampling data, non-ensemble , *ensemble bagging* (estimator yang sama) *dan ensemble voting* (3 estimator berbeda).

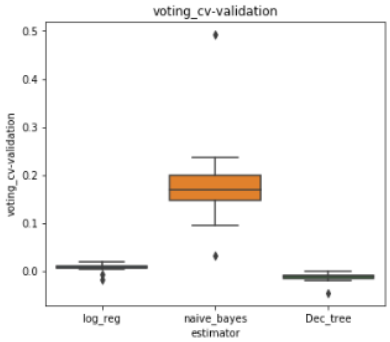
4. EVALUASI



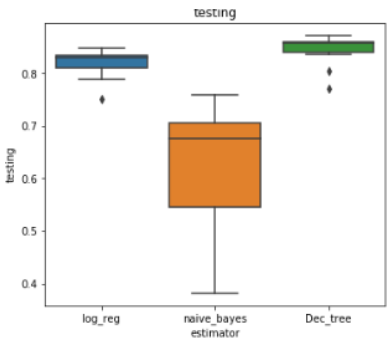
Gambar 9. Rata rata performansi validasi



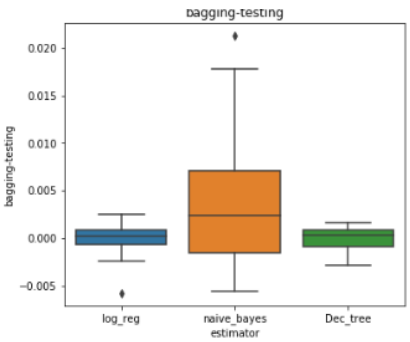
Gambar 10. Delta performansi validasi bagging



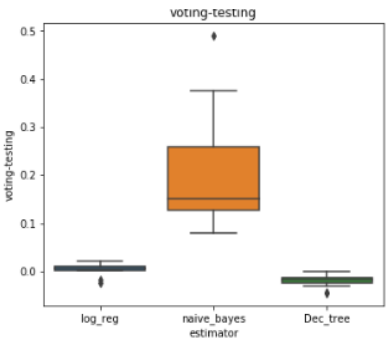
Gambar 11 Delta performansi validasi voting



Gambar 12. Rata rata performansi testing



Gambar 13. Delta performansi testing bagging



Gambar 14. Delta performansi testing voting

Dari Gambar 9 dan 12 dapat terlihat performansi validasi dan testing dari ketiga estimator. Decision tree memiliki performansi terbaik, logistic regressi memiliki performansi yang stabil dan naïve bayes memiliki rentangan quartal yang paling luas. Naïve bayes memiliki performansi tersebut dikarenakan naïve bayes sangat sensitive dengan data dengan jumlah data yang tidak imbang, sehingga data non sampling dan sampling memiliki performansi yang sangat jauh berbeda.

Delta performansi bagging pada Gambar 10 dan 13 terlihat bahwa naïve bayes terhadap *imbalance data* cukup terbantu. Dan menghasilkan median positive dan. Logistic regresi masih tergolong stabil dengan median positive dan decision tree memiliki performansi yang condong kearah negative.

Jike dilihat dari segi voting yaitu pada gambae 11 da 14, naïve bayes memiliki performansi positive yang cukup baik, ini terjadi dikarenakan estimator lain membantu saat naïve bayes tidak dapat menghasilkan performansi yang baik di *imbalance data*. decision tree memiliki performansi yang berbanding terbaik, dikarenakan terbebankan oleh kelemahal naïve bayes terhadap *imbalance data*. Dan logistic regresi masih sama yaitu relative stabil kearah positive.

5. KESIMPULAN

Sensitifitas Naïve Bayes terhadap data tak imbang terbantu oleh bagging dan voting dengan delta performansi 0.0001 – 0.0018, logistic regresi memiliki kenaikan performansi relative rendah untuk bagging dan voting dengan delta performansi 0.0001-0.00015, dan voting decision tree memiliki performansi yang terbebankan oleh Naïve Bayes dengan delta performansi -0.01.

Dari pengamatan hasil ensemble terhadap 3 estimator yaitu Naïve Bayes, Logistik Regresi dan Decision Tree,dengan 3 sampling yaitu non_sampling , random over sampling dan Kmean-Smote over sampling dapat disimpulkan bahwa ensemble bagging dan voting tidak selalu menghasilkan performansi lebih baik jika dilihat dari macro average f1 score terhadap dataset Email yang diperoleh dari Gmail tersebut.

Hal tersebut mungkin terjadi dikarenakan terlalu banyak data kotor atau salah label pada dataset, ada baiknya untuk penelitian selanjutnya menggunakan dataset seperti data IMDB, Movielens atau data lainnya yang sering digunakan dan diakui secara global dan mengganti naïve bayes dengan estimator probabilistic dan cukup kuat terhadap data tak imbang agar tidak ada estimator lain yang terbebankan saat voting.

DAFTAR PUSTAKA

[1] Wang. Xiao-lin, Cloete. Ian, “Learning to classify email: a survey”, 2015

[2] The Radicati Group.inc, “Email Statistics Report, 2018-2022”, 2018

[3] Tsugawa . Sho, Takahashi.Kazuya, Ohsaki. Hiroyuki , Imase. Makoto, “Robust Estimation of Message Importance using Inferred Inter-Recipient Trust for Supporting Email Triage”, 2010

[4] Singh. Bharat, Kushwaha. Nidhi, Vyas. Om Prkash, “A Scalable Hybrid Ensemble Model for Text Classification”, 2016

[5] Rong. Tongwen, Tian. Xing, Wing, “Location Bagging-based Undersampling for Imbalanced Classification Problems”, 2016

[6] Babar. Varsha, Ade. Roshani, “MLP-Based Undersampling Technique for Imbalanced Learning”, 2016

[7] L. Breiman, “Bagging predictors”, Machine Learning, 24(2), 123-140, 1996.

[8] V. Metsis, I. Androutsopoulos and G. Paliouras (2006). Spam filtering with Naive Bayes – Which Naive Bayes? 3rd Conf. on Email and Anti-Spam (CEAS)

[9] M. Dumont et al, Fast multi-class image annotation with random subwindows and multiple output randomized trees, International Conference on Computer Vision Theory and Applications 2009

[10] Hsiang-Fu Yu, Fang-Lan Huang, Chih-Jen Lin (2011). Dual coordinate descent. methods for logistic regression and maximum entropy models. Machine Learning 85(1-2):41-75. https://www.csie.ntu.edu.tw/~cjlin/papers/maxent_dual.pdf

[11] N. V. Chawla, K. W. Bowyer, L. O.Hall, W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” Journal of artificial intelligence research, 321-357, 2002.