ALGORITMA SUPER LEARNER SEBAGAI METODE ENSEMBLE DALAM PEMODELAN KLASIFIKASI EMAIL SPAM

LAPORAN DATA MINING
JOINTS 2020

Tim Agriminer V6.0

IPB University
Bogor Indonesia

Disusun oleh :

Julius Christoper Siswanto
Agus Hendra Nasution
Purnama Sari

IPB UNIVERSITY
BOGOR
2020

BAB I
LATAR BELAKANG

Lompatan kemajuan teknologi saat ini semakin pesat, baik teknologi komunikasi maupun informasi. Hampir semua hal dapat diintegrasikan dengan jaringan internet atau bisa dikenal "*Internet of things*" (IoT). Hal ini ditandai dengan adanya sistem otomatisasi hampir di semua bidang, bahkan menghapus ruang dan waktu. Salah satu contohnya adalah penggunaan *email*. Saat ini, *email* menjadi media komunikasi yang banyak digunakan karena jaminan efektif dan cepat yang diberikan. Ini memungkinkan pengguna untuk saling transfer informasi secara global dari mana saja dan kapan saja dalam sepersekian detik. Tidak hanya dalam komunikasi profesional dan bisnis, sebagian orang bahkan masih menggunakan *email* untuk komunikasi pribadi. Namun, seiring dengan peningkatan pengguna *email*, tak jarang *email* berisi konten tidak penting masuk dari berbagai sumber tidak dikenal. Inilah yang biasa disebut sebagai spam *email*.

Spam *email* adalah praktik mengirimkan data yang tidak diinginkan atau data massal ke beberapa akun *email*. Kegiatan spam *email* merupakan bagian dari spam elektronik dengan menggunakan pesan yang hampir identik dikirim ke berbagai penerima melalui *email*. Hal ini mulai berkembang sejak awal 1990-an. Biasanya, *spammer* mengumpulkan alamat *email* dari *chatroom*, situs web, daftar pelanggan, *newsgroup*, dan berbagai sumber lain. Sebuah *email* biasanya disebut spam jika memenuhi kriteria berikut. Pertama, alamat dan identitas pengirim *email* disembunyikan (*anonymity*). Kedua, *email* dikirim ke sekelompok besar orang (*mass mailing*). Ketiga, *email* tersebut tidak diminta oleh penerima (*unsolicited*). *Email* spam ini mempunyai tipe beragam. Ada yang berisi iklan atau promosi produk dan jasa tertentu, ada juga yang bertujuan untuk menyebarkan virus, *malware*, dan perangkat berbahaya lainnya. Tipe lain misalnya *phishing*, yaitu bersembunyi dibalik nama instansi dan lembaga tertentu dengan tujuan untuk mencuri informasi penting.

Banyaknya spam *email* saat ini tentunya menjadi tekanan bagi organisasi berbasis teknologi informasi dan menciptakan miliaran dolar kerugian dalam hal produktivitas. Selain itu, spam *email* juga menimbulkan pemborosan penyimpanan dan *bandwidth*, perusakan dan pencurian data penting, bahkan dengan spam *email* bisa juga terjadi penipuan dan tindak kriminal . Oleh karena itu, identifikasi *email* spam dan bukan spam menjadi bahan penelitian yang penting dan menarik guna meminimalisir berbagai bentuk kejahatan yang mungkin terjadi akibat adanya spam *email*.

BAB II
TUJUAN DAN MANFAAT

Tujuan :

1. Mengidentifikasi fitur dari *bag of word* yang memiliki kemampuan prediktif mengenai probabilitas suatu *email* spam.
2. Membangun model yang dapat memprediksi suatu *email* adalah spam .
3. Memahami jika satu model prediksi cocok untuk semua bahasa atau jika model spesifik harus dibangun untuk setiap bahasa berbeda.

Manfaat:

1. Mengetahui pola kata dari pesan *email* sehingga dapat dilakukan pemodelan yang bisa membantu pengguna untuk mengetahui *email* mana saja yang berpotensi spam.
2. Sebagai bentuk media pembelajaran dalam melakukan pemodelan dari sebuah *text mining*.

BAB III
METODE DATA MINING

1. Python

   Python adalah bahasa pemrograman interpretatif multiguna yang memakai filosofi perancangan dengan fokus kepada tingkat keterbacaan kode. Sebagai bahasa pemrograman, Python menggabungkan kemampuan, kapabilitas dan sintaksis kode serta fungsi pustaka yang berkualitas tinggi. Pada dasarnya, bahasa pemrograman yang dirancang oleh Guido Van Rossum ini sebenarnya sangat banyak digunakan untuk membuat program yang sangat sering dipakai oleh masyarakat umum. Sebut saja program GUI (desktop), aplikasi *smartphone*, program CLI, Data Mining, IoT, *game*, web, program untuk *hacking* dan masih banyak lagi.

2. Dataset

   Pada penelitian ini, kami menggunakan dataset "*Email Spam Classification*" dari kasus banyak email spam yang diterima mbak Mawar. Dataset ini memiliki 42 feature (atribut atau variabel) dan 5172 observasi. Masing - masing *feature* terdiri atas *bag of word* yang berupa frekuensi banyaknya kata yang muncul dalam satu pesan *email*.

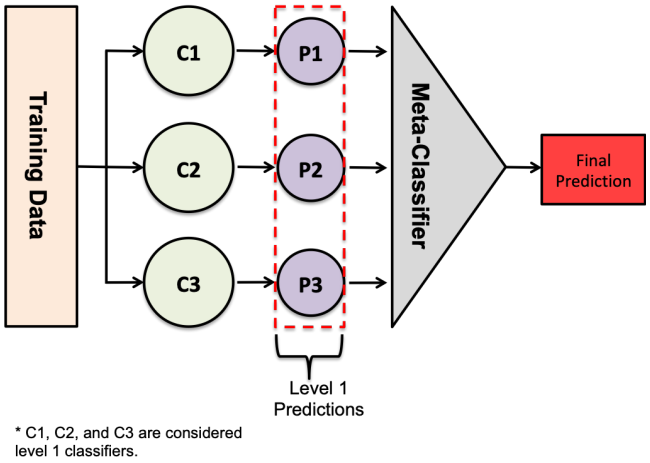
3. Klasifikasi

   Klasifikasi pada dasarnya merupakan proses menilai objek dan pengelompokan peubah ke dalam kelas yang sudah ditentukan (Linoff dan Berry 2011). Target yang akan diprediksi dalam klasifikasi merupakan peubah yang bersifat kategorik. Metode klasifikasi dibedakan ke dalam dua kelompok, yaitu non parametrik dan parametrik. Algoritma klasifikasi non parametrik tidak membutuhkan asumsi sebaran. Ada berbagai macam teknik klasifikasi yang termasuk ke dalam kelompok non-parametrik, antara lain CART, *neural network*, *support vector machine*, dan *random forest*, sedangkan yang termasuk ke dalam kelompok parametrik antara lain regresi logistik dan analisis diskriminan. Terdapat dua tahapan atau proses dalam melakukan klasifikasi, yaitu proses pelatihan dan proses pengujian. Pada proses pelatihan model dibangun dengan menggunakan data latih yang telah diketahui labelnya, sedangkan proses pengujian dilakukan untuk mengetahui kinerja model yang telah dibangun dengan menggunakan data uji.

4. Super Learner

   Stacked generalization adalah sebuah metode *ensemble* yang memungkinkan peneliti untuk menggabungkan beberapa algoritma yang berbeda menjadi 1 model. Semenjak diperkenalkan di awal tahun 1990-an, metode ini telah berkembang sesuai perkembangan zaman menjadi yang dikenal sebagai *Super Learner*. *Super Learner* menggunakan *V-fold cross-validation* untuk membuat kombinasi pembobot yang paling

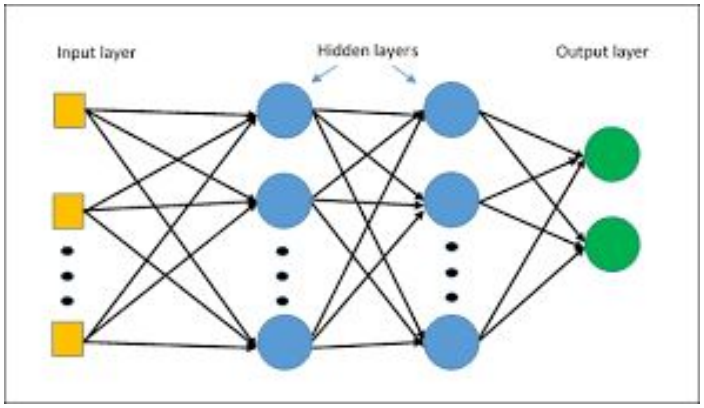
optimal untuk setiap kandidat algoritma. Optimasi ditentukan oleh fungsi objektif yang ditentukan pengguna, seperti meminimalkan kesalahan kuadrat rata-rata atau memaksimalkan area di bawah kurva ROC (Naimi, 2017).



Gambar 1 Algoritma *Super Learner*

5. Multi Layer Perceptron

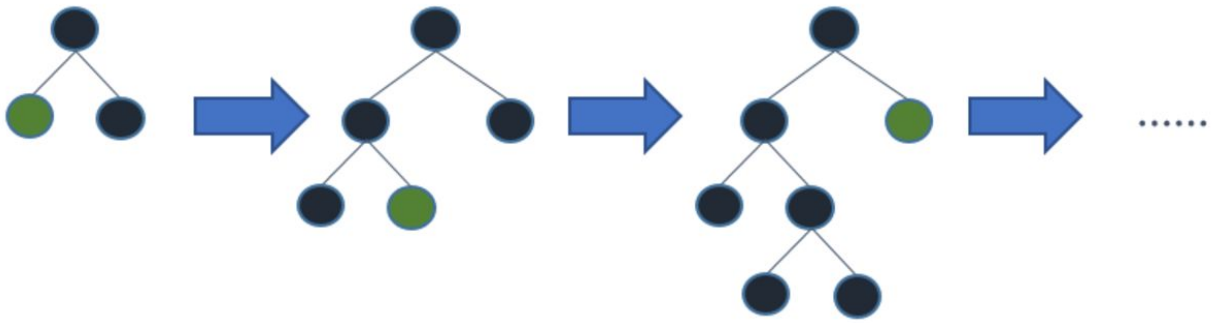
Metode klasifikasi *Multilayer Perceptron* (MLP) merupakan salah satu jenis dari algoritma jaringan syaraf tiruan yang mengadopsi cara kerja jaringan saraf pada makhluk hidup. Algoritma ini terkenal handal karena proses pembelajaran yang mampu dilakukan secara terarah. Pembelajaran algoritma ini dilakukan dengan memperbarui bobot balik (*backpropagation*). Penetapan bobot yang optimal akan menghasilkan hasil klasifikasi yang tepat. MLP terdiri dari sistem yang sederhana saling menghubungkan jaringan atau node yang diilustrasikan pada gambar dibawah. Node tersebut dihubungkan oleh bobot dan unit *output* yang merupakan fungsi penjumlahan dari *input* ke node dimodifikasi oleh transfer non-linear sederhana, atau aktivasi.



Gambar 2 *Multi Layer Perceptron*

6. LightGBM

Light GBM adalah salah satu algoritma *machine learning* yang menggunakan algoritma pembelajaran berbasis pohon. Pada konsepnya LightGBM bekerja dengan membangun pohon secara vertikal, sementara algoritma lain membangun pohon secara horizontal. Artinya, LightGBM meminimalkan bias sementara algoritma lain meminimalkan *variance*. Ini akan memilih daun dengan *max delta loss* untuk tumbuh.



Leaf-wise tree growth

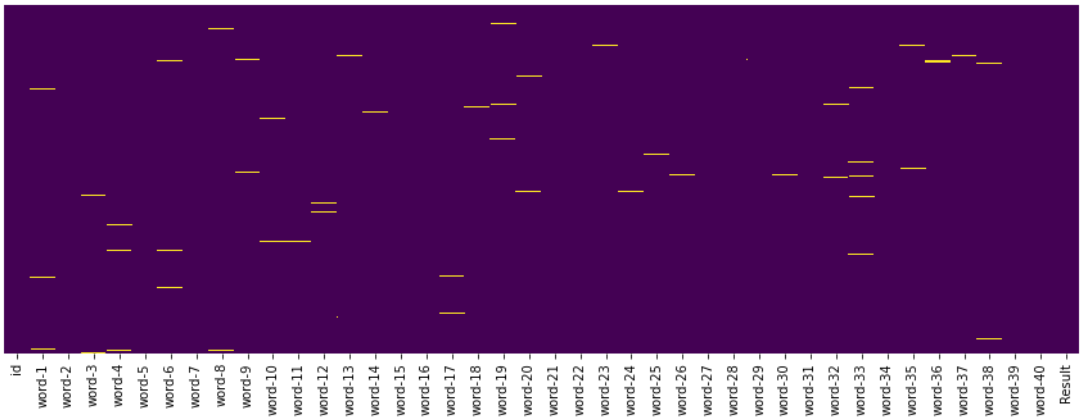
7. ROC *Curve*

Kurva ROC adalah tampilan grafis sensitivitas (TPR) pada sumbu y dan (1 -spesivisitas) (FPR) pada sumbu x untuk berbagai titik batas nilai tes. Area di bawah kurva (*Area Under Curve* / AUC) adalah ukuran sensitivitas yang efektif dan kombinasi dan spesifisitas untuk menilai validitas inheren atas diagnostik. Nilai maksimum AUC adalah 1, menandakan tes diagnostik sangat sempurna dalam membedakan sebuah pesan *email* masuk ke dalam kategori spam atau tidak.

BAB IV
ANALISIS

A. Pra Proses Data dan *Feature Engineering*

Pada penelitian ini, kami menggunakan dataset "*Email Spam Classification*" dari kasus banyak email spam yang diterima mbak mawar. Dataset ini memiliki 42 *feature* (atribut atau variabel) dan 5172 observasi. Masing - masing *feature* terdiri atas *bag of word* yang berupa frekuensi banyaknya kata yang muncul dalam satu pesan *email*. Perubahan peubah dengan tipe data *object* dilakukan dengan menghilangkan karakter yang tidak berupa angka agar tidak menjadi *missing value* saat konversi tipe data menjadi *integer*. Ringkasan banyaknya *missing value* dari tiap peubah di dalam data *email spam classification* ditampilkan pada gambar 3.



Gambar 3 Visualisasi *missing value* pada data

Pengisian *missing value* dilakukan berdasarkan skala pengukuran dari peubahnya. Untuk data dengan skala pengukuran numerik, maka dilakukan imputasi dengan menggunakan median. Untuk meningkatkan akurasi pada proses klasifikasi, salah satu cara yang dapat dilakukan adalah dengan menambahkan peubah baru ke dalam data berdasarkan peubah yang ada. Informasi mengenai peubah baru tersebut dapat dilihat pada tabel 1.

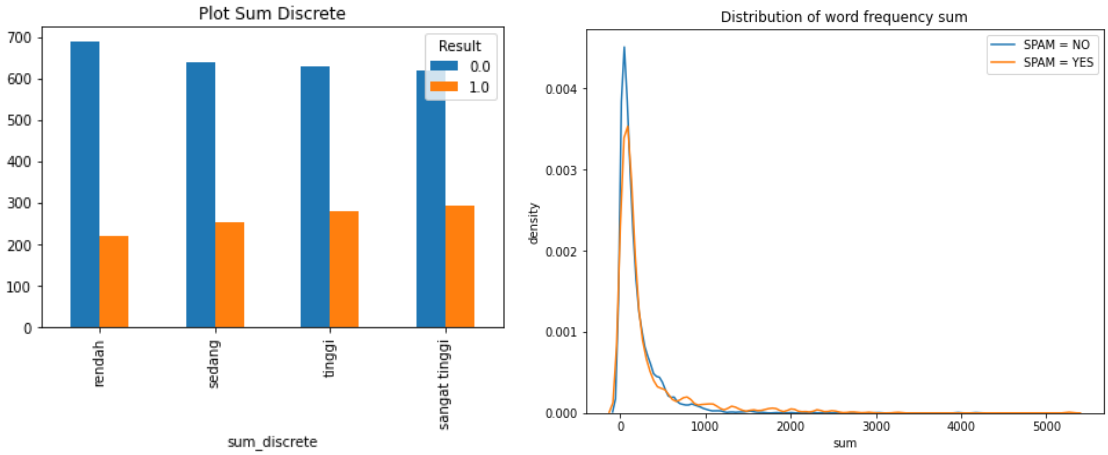
Tabel 2 Atribut baru

No	Nama Peubah	Penjelasan	Skala
1	no_missing	Banyaknya jenis kata yang muncul dibagi 40	Boolean
2	sum	Total frekuensi dalam satu email	Numeric
3	word_max	Frekuensi tertinggi	Numeric

4	word_min	Frekuensi terendah	Numeric
5	range	Selisih antara word_max dengan word_min	Numeric
6	count_0	Banyaknya kata yang memiliki nilai (dalam satu email	Numeric
7	abs_avg	Rata - rata dari word_max dengan word_min	Float

B. Eksplorasi dan Analisis Data

1. Jumlah Frekuensi Kata

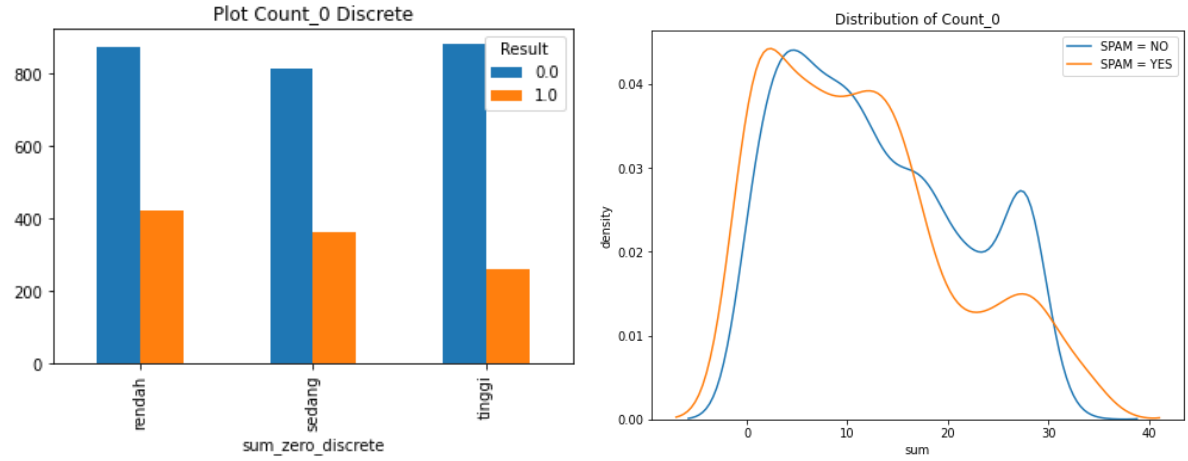


Gambar 4 Visualisasi jumlah frekuensi kata

Saat *pre-processing* data, dibentuk sebuah peubah baru yang merupakan penjumlahan frekuensi dari masing-masing baris. Setelah itu, dilakukan diskretisasi hasil menjadi rendah, sedang, tinggi, dan sangat tinggi seperti pada gambar 2. Kata yang termasuk spam pada email diberi kode 1, sedangkan bukan spam diberi kode 0. Pada grafik sebaran jumlah frekuensi kata, dapat dilihat dengan jelas bahwa kata yang berjenis spam lebih sedikit dibandingkan dengan kata bukan spam pada sebuah *email*.

Hasil diskretisasi diatas menunjukan bahwa semakin tinggi tingkat frekuensi suatu kata muncul pada satu email maka semakin yakin juga email itu masuk kedalam kelas spam. Pada *density plot* terlihat bahwa kelas spam dan tidak spam memiliki pola sebaran yang sama yaitu menjulur ke kanan, hal ini dikarenakan adanya pencilan yang ada pada beberapa kata.

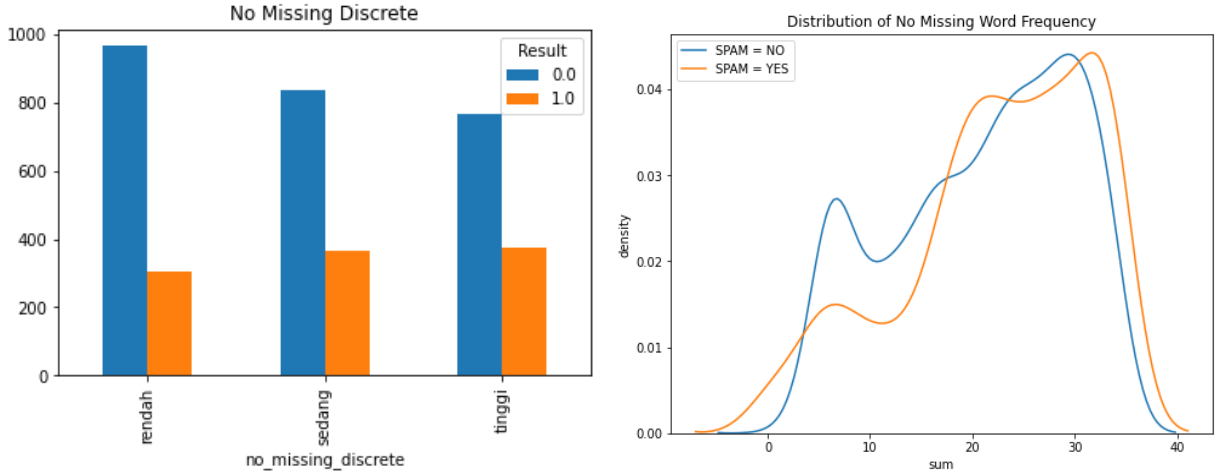
2. Jumlah Kata Bernilai 0



Gambar 5 Visualisasi jumlah kata bernilai nol

Dari 40 jenis kata yang ada, terdapat beberapa kata yang tidak muncul (frekuensi kemunculan nol). Hasil yang diperoleh seperti pada gambar 3. Jika jenis kata yang bernilai 0 cukup tinggi maka peluang suatu email masuk ke dalam kelas spam cukup rendah, begitu juga sebaliknya. Kata yang paling banyak mengandung angka 0 terdapat pada word-0 sampai word-10, hal ini mengindikasikan bahwa *word* kurang dari *word* -10 tidak terlalu sering digunakan dalam melakukan aktivitas spam.

3. Banyaknya jenis kata

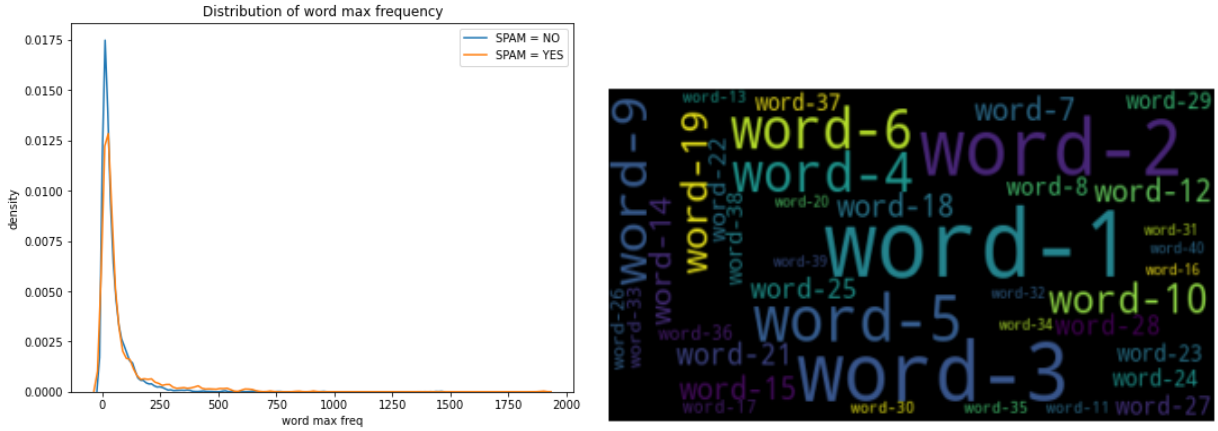


Gambar 6 Visualisasi jenis kata

Atribut lain yang dibentuk adalah no_missing, yaitu atribut yang berisi jenis kata yang memiliki frekuensi bernilai tidak nol. Banyaknya jenis kata pada *email* ditampilkan dalam bentuk bar plot seperti pada gambar 4. Terdapat lebih dari 900 email memiliki

jenis kata yang hilang sedikit, maka peluang masuk ke dalam kategori tidak spam semakin tinggi. Namun semakin tinggi email yang kehilangan jenis kata maka peluang masuk ke dalam kelas spam tinggi. Dilihat dari *density plot* menunjukan bahwa jenis kata yang paling sering digunakan adalah *word*-30 sampai *word*-40. Artinya jika penggunaan kata ini sangat banyak sedangkan jenis kata yang digunakan sedikit, maka bisa dikategorikan sebagai *email* spam.

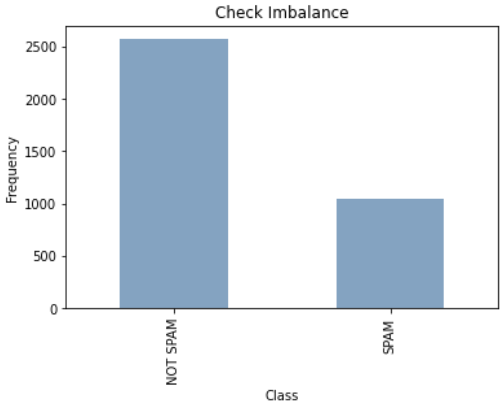
4. Frekuensi kata tertinggi pada setiap email



Gambar 7 Visualisasi jumlah kata maksimum

Dari beberapa jenis kata yang ada, terlihat bahwa *word* -1 memiliki frekuensi kemunculan tertinggi pada sebuah email. *Word cloud* yang terbentuk menggambarkan jenis kata yang paling banyak muncul pada setiap *email*. Semakin besar ukuran dari masing-masing kata, menunjukkan frekuensi kemunculan yang semakin besar pula. Pada gambar 8, *word* - 1 memiliki frekuensi kemunculan terbesar pada *email* dan selanjutnya diikuti *word*-3 dan *word*-2 dengan frekuensi kemunculan besar.

5. Data *Imbalance*



Gambar 8 Visualisasi data *imbalance*

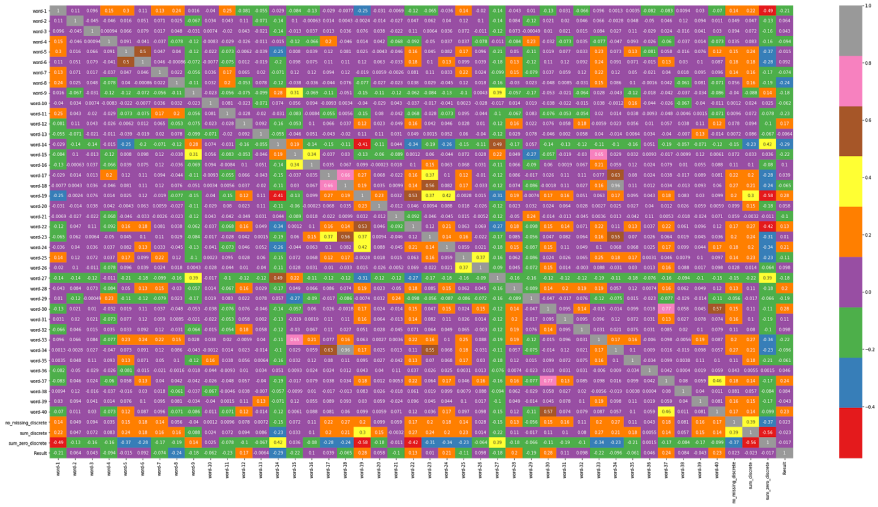
Plot diatas menunjukan respon dari data yang *imbalance*, sebanyak 31.5% data masuk kedalam kategori spam dan 68.5% masuk ke dalam kelas tidak spam. Hal mengindikasikan harus adanya perlakuan khusus terhadap data sebelum dilakukan pemodelan, *stratify* adalah teknik yang penulis gunakan dalam mengatasi permasalahan ini.

C. Anomali Data

Saat melakukan eksplorasi terhadap data, penulis melihat adanya beberapa *outlier* ataupun *noise* pada beberapa peubah (yang selanjutnya akan dibuang demi kebaikan model), seperti :

- Data yang bernilai nol, bahkan ada yang negatif yaitu pada atribut *word*-4 dan *word*-25.
- Ada sekitar dua data *outlier* di mana nilai dari atribut *word*-25 lebih dari 50000. Ada juga beberapa data outlier dimana nilai atribut *word*-1 lebih dari 10000.
- Adanya data *noise* dimana nilai dari jenis kata sedikit tetapi memiliki frekuensi yang sangat tinggi, maka sudah pasti *email* spam.
- Adanya data yang duplikat, hal ini menunjukan bahwa kemungkinan pesan yang sama tidak ditujukan hanya ke satu pengguna melainkan banyak orang.

D. Matriks Korelasi



Gambar 9 Visualisasi korelasi antar peubah

Setelah melewati tahap pra proses, kita akan melihat matriks korelasi untuk melihat hubungan dari masing - masing *feature*. Berdasarkan matriks korelasi di atas, setidaknya ada 9 warna yang menandakan hubungan antara atribut baik positif maupun negatif. Untuk atribut yang memiliki warna terang mengindikasikan hubungan positif, dan untuk atribut yang memiliki warna gelap mengindikasikan hubungan yang negatif.

E. Pencocokan Model

Proses klasifikasi "*Email Spam Classification*" dicoba dengan menggunakan beberapa metode, seperti *AdaBoost, Random Forest, GradientBoost, LightGBM*, *BalancedBagging* dengan *base learner LightGBM*, MLP, dan *VotingClassifier* untuk semua data. Tentu saja data yang akan diklasifikasikan harus sudah dirapikan terlebih dahulu seperti, mengubah tipe data *string* menjadi *numeric*, serta membuat *feature engineering*. Selanjutnya, dilakukan pemisahan data latih dan data uji sebesar 70% untuk data latih dan 30% untuk data uji. Berikut hasil prediksi dari masing - masing model.

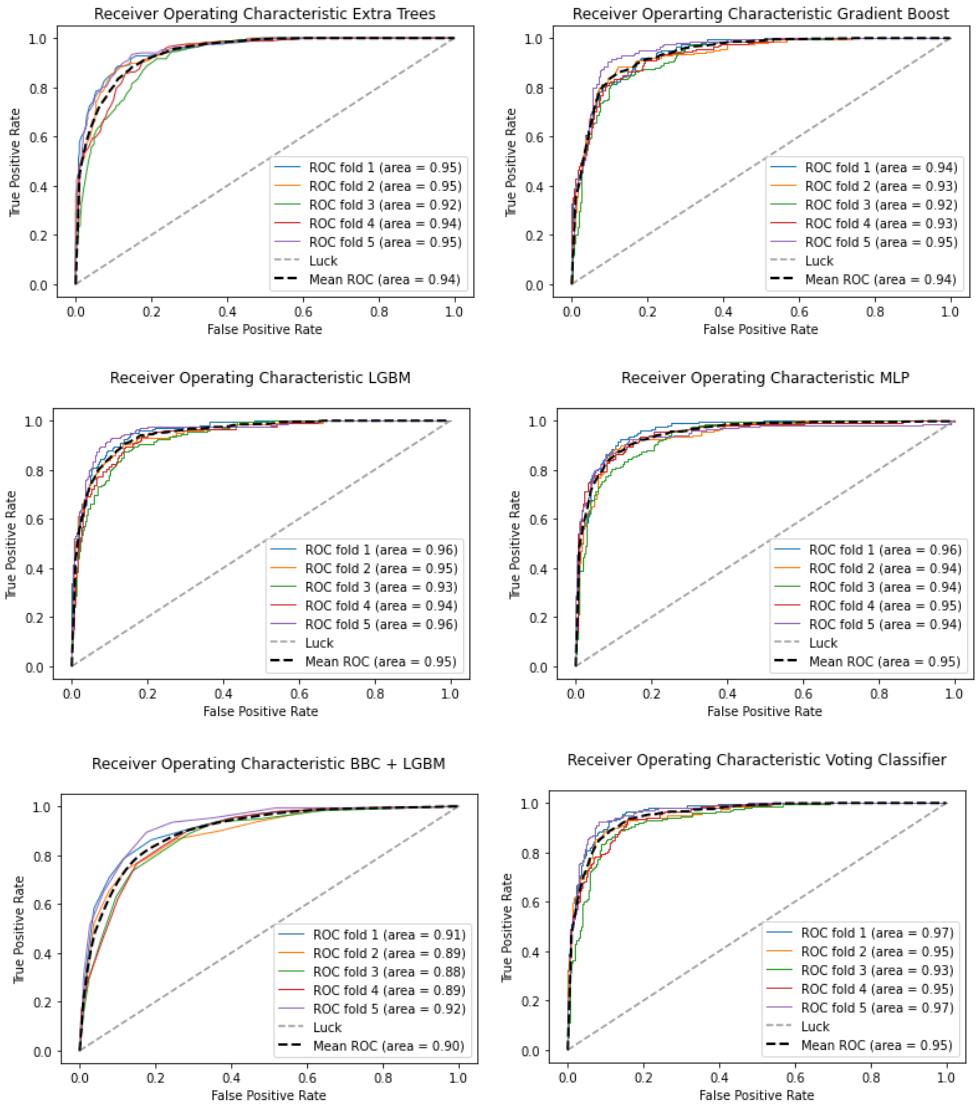
Tabel 3 *Result set* algoritma

Model	Accuracy	Precision	Recall	F1-Score	ROC
AdaBoost	0.8782	0.8666	0.6842	0.8413	0.94
GBoost	0.8713	0.7989	0.7783	0.8561	0.94
LGBM	0.8921	0.8078	0.8453	0.8802	0.95
Random Forest	0.8824	0.8114	0.7319	0.8453	0.94
Balanced Bagging	0.8395	0.7470	0.8807	0.8573	0.90
MLP	0.8893	0.8685	0.7272	0.8581	0.95
Voting Classifier	0.8934	0.8333	0.7894	0.8828	0.95

Tabel 3 menunjukkan hasil akurasi, presisi, *recall*, dan ROC dengan *hold-out method*. Tiga metode yang memiliki tingkat akurasi dan ROC tertinggi yakni *LightGBM*, *MLP*, dan *Voting*.

F. Evaluasi

Untuk mengevaluasi model penulis membandingkan hasil nilai akurasi dan ROC yang menggunakan *hold-out method* dengan *stratified cross validation*. Berikut adalah hasil akurasi dan ROC antara *hold out* dan *cross validation* :



Gambar 10 Kurva ROC

Tabel 4 *Result set* algoritma

Model	Accuracy	Cross Validation
AdaBoost	0.8675	0.8654
GBoost	0.8713	0.8706
LGBM	0.8921	0.8948
Random Forest	0.8824	0.8793
Balanced Bagging	0.8395	0.8741

MLP	0.8893	0.8582
VotingClassifier	0.8934	0.8941

Hasil akurasi pada setiap metode dengan *stratified cross validation* lebih tinggi dibandingkan dengan *hold-out method*. Kecuali untuk model *Random Forest*, *Gradient Boosting, BalancedBagging,* dan *VotingClassifier*. Sedangkan untuk ukuran nilai kebaikan model bisa dilihat pada tabel dibawah ini.

Tabel 5 Perbandingan nilai *error*

Model	MAE	MSE	RMSE
AdaBoost	0.1244	0.1244	0.3528
GBoost	0.1120	0.1120	0.3347
LGBM	0.0954	0.0954	0.3089
Random Forest	0.1244	0.1244	0.3528
Balanced Bagging	0.1175	0.1175	0.3428
MLP	0.1203	0.1203	0.3468
VotingClassifier	0.1065	0.1065	0.3263

G. Hyperparameter Tuning

Hyperparameter Tuning penulis gunakan untuk semua model yaitu *BalancedBagging, LightGBM, MLP, RandomForest, GradientBoost,* dan *AdaBoost* berdasarkan akurasi. Berikut adalah *hyperparameter tuning* pada masing-masing metode.

1. VotingClassifier

   Masing-masing model yang digunakan dalam *VotingClassifier* menggunakan parameter terbaik berdasarkan akurasi dengan *Grid Search Cross Validation Hyperparameter Tuning.* Terdapat 6 model yang akan dimasukkan ke dalam *VotingClassifier* yaitu *BalancedBagging, LightGBM, MLP, RandomForest, GradientBoost,* dan *AdaBoost.*

2. Balanced Bagging

Parameter terbaik untuk metode *Bagging* adalah *criterion* dengan *base_estimator* menggunakan *LightGBM*, *n_estimators* sebesar 140, dan *random_state* sebesar 12, *Best Score* yang diperoleh sebesar 0.90.

3. LightGBM

   Parameter terbaik untuk metode *LightGBM* adalah *n_estimators* sebesar 1000, *random_state* sebesar 314, *max_depth* adalah -1, *silent* berupa *true metric* adalah *None*, dan *n_jobs* adalah 4. *Best score* yang diperoleh untuk *LightGBM* sebesar 0.90.

4. AdaBoost

   Parameter terbaik untuk metode *AdaBoost* adalah *n_estimators* sebesar 2, *random_state* sebesar 7, *max_feature* sebesar 10, *bootstrap False*, nilai *criterion gini*, dengan *base estimator random forest classifier*. *Best score* untuk *AdaBoost* sebesar 0.88.

5. Multilayer Perceptron (MLP)

   Parameter terbaik untuk metode MLP adalah *activation* berupa tanh, nilai alpha 0.0001, beta sebesar 0.9, nilai epsilon sebesar 1e-08, *learning_rate adaptive*, *max_fun* 15000, *max_iter* 200, dan nilai *tol* sebesar 0.0001. *Best score* untuk MLP sebesar 0.89.

6. Random Forest

   Parameter terbaik untuk metode *RandomForest* adalah dengan tidak menggunakan *bootstrap*, menggunakan *gini* sebagai kriteria pembelajaran, *max_features* 10, *min_samples_leaf* 1, *min_samples_split* 3, dan *n_estimators* 300. *Best score* untuk *Random Forest* sebesar 0.88

7. Gradient Boost

   Parameter terbaik untuk metode *GradientBoost* adalah dengan menggunakan *friedman_mse criterion*, *learning_rate* =0.1,*max_depth* 8, *max_features* 0.3, *min_sample_leaf* = 100, *min_samples_split* =2 dan *n_estimators* =300. *Best score* untuk *Gradient Boost* sebesar 0.86

H. Model Terbaik

Nilai akurasi, presisi, recall, dan ROC pada model setelah dilakukan *hyperparameter tuning* adalah sebagai berikut :

Tabel 6 *Result set* algoritma

Model	Accuracy	Precision	Recall	F1-Score	ROC
AdaBoost	0.8796	0.8546	0.7033	0.8449	0.94
GBoost	0.9004	0.8407	0.8086	0.8774	0.94
LGBM	0.9017	0.8484	0.8038	0.8786	0.95
Random Forest	0.8852	0.8387	0.7464	0.8554	0.94
Balanced Bagging	0.8796	0.7459	0.8851	0.8608	0.95
MLP	0.8796	0.8708	0.7520	0.8598	0.95
Voting Classifier	0.9128	0.865	0.8277	0.8926	0.96

Tabel 7 *Result set* algoritma

Model	Accuracy	Cross Validation
AdaBoost	0.8796	0.8803
GBoost	0.9004	0.8897
LGBM	0.9017	0.9028
Random Forest	0.8852	0.8793
BalancedBagging	0.8796	0.8806
MLP	0.8796	0.8817
Voting Classifier	0.9128	0.9038

Tabel 8 Perbandingan nilai *error*

Model	MAE	MSE	RMSE
AdaBoost	0.1203	0.1203	0.3468
GBoost	0.099	0.099	0.3155
LGBM	0.0982	0.0982	0.3133

Random Forest	0.1092	0.1092	0.3305
Balanced Bagging	0.1203	0.1203	0.3468
MLP	0.1106	0.1106	0.3326
Voting Classifier	0.0871	0.0871	0.2951

Nilai akurasi, presisi, *recall*, F1-*score*, serta ROC pada metode diatas sangat besar. Hasil dari akurasi dan F1-*score* diatas 89% untuk VotingClassifier, artinya nilai data yang diujikan cukup baik untuk memprediksi data aktual yang *imbalance*, serta tingkat keberhasilan sistem dalam menemukan sebuah informasi juga cukup baik, dapat dilihat dari nilai *recall* VotingClassifier diatas 82%. Baik nilai akurasi, *recall*, presisi, F1-*score* maupun ROC yang tinggi dapat menjadi hal baik, namun patut dikoreksi dan ditinjau kembali pada bagian pengolahan data karena nilai *recall* dari masing-masing model mayoritas di bawah 81%. Penulis telah mencoba metode-metode klasifikasi tersebut dengan data yang sudah diseimbangkan atau data yang *balance*. Nilai akurasi yang dihasilkan meningkat signifikan untuk model VotingClassifier. Sehingga, dapat diselidiki lebih lanjut mengenai apa yang menyebabkan besarnya tingkat *recall* pada penelitian ini kecil. Setelah dilakukan *hyperparameter tuning* didapatkan nilai galat pada VotingClassifier adalah yang terkecil dengan nilai MAE, MSE, dan RMSE sebesar 0.0871, 0.0871, dan 0.2951. Hal ini menjadikan VotingClassifier sebagai model yang tepat untuk penelitian ini.

BAB V
KESIMPULAN

Dari hasil dan pembahasan yang telah dilakukan dalam prediksi dan ciri - ciri *email* spam didapat bahwa sebanyak 10 fitur memiliki kemampuan prediktif mengenai probabilitas *email* spam dengan keakuratan yang sangat baik dan tidak terjadi *overfitting* pada model VotingClassifier. Untuk rata-rata ukuran akurasi, presisi, *recall*, dan *f1-score* dari keseluruhan gugus data mengalami kenaikan walaupun tidak terlalu besar setelah dilakukan *hyperparameter tuning*. *Feature* baru yang diyakini berpengaruh untuk prediksi *email* yaitu *count_0* dan no_missing. Model VotingClassifier dengan model *BalancedBagging, LightGBM, MLP, RandomForest, GradientBoost,* dan *AdaBoost* diketahui memiliki nilai *accuracy, recall*, presisi dan F1-*score* paling tinggi dibandingkan dengan model lain, hal ini bisa dianggap bahwa model tersebut cukup baik untuk mengklasifikasi *email* spam.

BAB VI
DAFTAR PUSTAKA

Dewaweb. 2019. *Keunggulan Memahami Bahasa Pemrograman Python* https://www.dewaweb.com/blog/keunggulan-memahami-bahasa-pemrograman-python/ [diakses 29 April 2020].

Ervita, K P, Tedy S. 2014. *Penerapan Text Mining pada Sistem Klasifikasi Email Spam Menggunakan Naïve Bayes.* Yogyakarta: Universitas Ahmad Dahlan.

Medium. 2017. *What is LightGBM, How to implement it? How to fine tune the parameters?*.https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc [diakses 3 Mei 2020]/

Mentari M, Sari E.K.R, Mutrofin S. *Klasifikasi Menggunakan Kombinasi Multilayer Perceptron dan Alignment Particle Swarm Optimization.* Bangkalan : SENASTIK.

Nadeak F. 2019. *Kajian Algoritme Super Learner sebagai Metode Ensemble dalam Pemodelan Klasifikasi.* Bogor: IPB University.

Naimi AI, Balzer L. 2017. *Stacked Generalization: An Introduction to Super Learning.* 35:3-9

Dokumentasi: