**Abstrak**

Short Message Service atau SMS adalah layanan umum yang mungkin untuk mengimkan dan menerima pesan didala bentuk teks terhadap perangkat seluler yaitu telefon genggam. Kalimat spam pada sms adalah sebuah pesan yang sangat tidak diinginkan oleh pengguna dalam menggunakan telefon genggam dan yang pengguna tidak inginkan tersebut berada pada kotak penerimaan pesan atau inbox message. Untuk mengatasi salah satu permasalahan ini adalah dengan cara melakukan mengklasifikasikan kalimat pesan SMS untuk ditentukan tipe spam menggunakan machine learning. Penelitian ini perlu melakukan percobaan untuk membandingkan tingkat akurasi terhadap penerapan metode supervised learning untuk membandingkan tingkat akurasi penerapan pada metode supervised learning menggunakan algoritma Naïve Bayes Classifier dalam melakukan klasifikasi data pada pesan SMS. Supervised learning adalah metode untuk mengklasifikasikan data berdasarkan label yang sudah ada. Naïve Bayes adalah metode klasifikasi pada data dengan berdasarkan nilai peluang. Numpy adalah metode yang bisa menggunakan operasi aritmatika dan menghitung rata-rata pada prediksi. TF-IDF adalah metode untuk melakukan pembobotan pada hubungan suatu kata atau text terhadap dokumen. Dengam menggabungkan metode ini dan menggabungkan tiga konsep untuk perhitungan bobot yaitu Term Frequency yang merupakan kemunculan kata pada kalimat, document frequency adalah jumlah banyak dokumen yang dimana suatu kata muncul. Dari hasil telah dilakukan terhadap eksperimen yang telah dilakukan, dengan menggunakan pembagian data 70% data training dan 30% data testing telah diperoleh terhap algoritma Naïve Bayes (NB) menghasilkan akurasi yang paling tinggi adalah 95,65% dibandingkan terhadap beberapa algoritma lain.

***Kata Kunci:*** *Spam SMS, Klasifikasi Spam SMS, Naïve Bayes, Numpy, TF-IDF*

**METODE PENELITIAN**

**Pengumpulan Dataset**

*Dataset* yang telah dikumpulkanyaitu diambil dari Crawling data dari twitter dan dari berbagai macam platform contohnya kaggel, twitter, github. Banyaknya jumlah data yang telah dikumpulkan dari berbagai repository publik yaitu adalah sebesar 1144 yang terdiri dari 334 kalimat pesan spam lalu terdapat 569 kalimat pesan normal dan 239 kalimat promo. Sedangkan untuk total data yang didapat dari Crawling data Twitter yaitu sebesar 3000 kalimat pesan dengan pesan normal senesar … kalimat dan …kalimat spam. Lalu setelah semua data yang telah didapat sudah terkumpul. Jika ditemukan tanda baca didalam dataset maka data tersebut juga akan dihapusjab. Sehingga semua data yang telah dikumpulkan dan dijadikan satu akan menjadi unik.

**Preprocessing Dataset**

Pada tahapan ini terdapat beberapa proses yang harus dilakukan yaitu *case folding,* dan *stopword.*

1. Stopword

Stopwords adalah kata-kata bahasa Inggris yang tidak menambah banyak arti kalimat. Mereka dapat dengan aman diabaikan tanpa mengorbankan makna kalimat. Misalnya kata-kata seperti the, he, have dll. Kata-kata tersebut sudah tercakup dalam corpus yang bernama corpus.

1. Stemming

Stemming adalah sebuah proses dalam mengembalikan setiap kata Kembali dalam bentuk dasarnya dengan menghilangkan semua akhiran yang terdapat dikata tersebut seperti -ing dan -er. Ketika dilihat dalam Bahasa inggris kata-kata yang memiliki akhiran adalah seperti , finisher, finishing. Lalu kata tersebut akan diubah menjadi satu kata yang sama.

**Preprocessing Data**

1. Naïve Bayes Classifier

Algoritma Naïve Bayes ini adalah metode baru yang digunakan untuk mengklasifikasikan sejumlah dokumen. Naïve Bayes ini memanfaatkan metode probabilitas dan statistic ang dikemukakan oleh ilmuan di negara inggris dengan nama Thomas Bayes.

Dasar dari torema rumus Naïve Bayes yang digunakan dalam pembahasan ini adalah sebagai berikut:

P(a|b)=(P(b|a) \* P (a)) / P (b))

Peluang kejadian a sebagai b akan ditentukan dari peluang b saat ampeluang a, dan peluang b. Dalam pengaplikasian rumus tersebut akan berubah menjadi

P(ci|d)=(P(d|c) \* P (ci)) / P (d))

Naïve Bayes Classifier atau dapat juga disebut dengan Multinominal Naïve Bayes merupakan model penyederhanaan dari algoritma bayes yang sesuai untuk pengklasifikasian text. Persamaan nya sebagai berikut:

Vmap = arg max P(vi | a1, a2, … an)

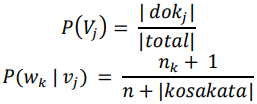
Pada persamaan di atas maka persamaan pertama dapat dituliskan dengan:



Didalam hal P(a1, a2, … an) konstan dapat dihilangkan, maka akan menjadi persamaan:



Penjelasan:



P (Vi) adalah probabilitas dari setiap dokumen terhadap sejumlah dokumen

P (Wk | adalah probabilitas kemuncukan kata wk pada suatu dokumen dengan kategori kelas Vi

Dok adalah frekuensi dokumen dari setiap kategori

Total adalah jumlah dokumen yang ada

Nk adalah frekuensi kata ke – k pada setiap kategori

|kosakata| : jumlah kata pada dokumen tes

1. TF-IDF

TF-IDF merupakan metode yang ter integrasi antar *term frequency* (TF), dan *inverse document frecuency* (IDF). Term Frequenc akan dihitung menggunakan persamaan dengan term frequency ke i yaitu frecuency kemunculan term ke I dalam dokumen je j inverse document frecuency (IDF) adalah algoritma dari rasio jumlah seluruh dokumen dalam korpus dengan sejumlah dokumen yang memiliki term dimaksud seperti yang ditulis pada matematis pada persamaan. Nilai yang didapat dengan mengkalikan kedua nya yang di informasikan pada persamaan.   

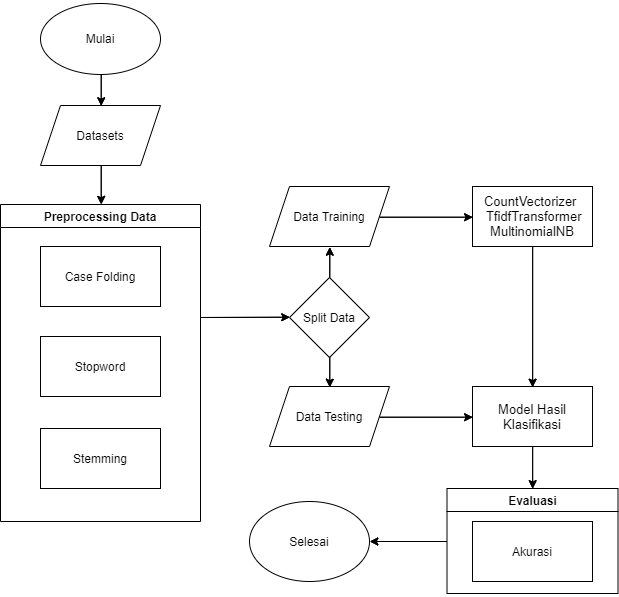
Kegunaan dari metode TF-IDF sendiri yaitu mencari representasi nilai dari setiap dokumen pada suatu sekumpulan data training yang dimana nantinya akan dibentuk suatu vektorantara dokumen dengan kata yang dimana untuk kesamaan antara dokumen dengan cluster yang di tentukan oleh sebuah prototype vector yang disebut juga dengan cluster centroid.

1. CountVectorizer

Model Regresi Logistik tercapai akurasi paling luar biasa di antara yang tersedia pilihan (95 persen). Saat datang ke untuk mengidentifikasi sampel, kami menggunakan enam yang diawasi teknik pembelajaran mesin: Logistik Regresi (dengan K-NN), Pohon Keputusan (dengan DT), SVC (dengan SVC), Naive Bayes (dengan Random Forest), dan Hutan Acak (dengan Random Forest).

**Perancangan Eksperimen**

Pada Perancangan disain eksperimen ini pengerjaan tugas internship 1 adalah sebagai berikut:



Mengenai penjelasan pada hasil rancangan eksperimen adalah sebagai berikut:

1. System akan menerima input file yang berupa datasets yang berekstensi CSV.
2. Setelah itu sebelum digunakan dataset akan melakukan tahap *preprocessing datasets* terlebih dahulu menggunakan *case folding, stopword* dan *stemming.*
3. Setelah datasets melewati tahap preprocessing, datasets tersebut akan di splitting.
4. Setelah selesai melewati tahap splitting maka akan dibagi menjadi dua yaitu data training dan testing. Data yang sudah menjadi data training dan testing akan berguna untuk melatih pada system untuk dapat mengklasifikasikan data pada kalimat SMS. Sedangkan data testing akan berguna untuk menguji keakuratan pada system yang telah dilatih sebelumnya. Pembagian data yang dilakukan adalah sebagai berikut:
5. Mengenai pembadian pertama yaitu data sebesar 80% data training dan 20% data testing.
6. Setelah mendapatkan model dari ketiga algoritma tersebut, dilakukan tahap evaluasi dengan menghitung akurasi dan juga error setelah itu kita akan menampilkan plot nya

**Sample Pengujian**

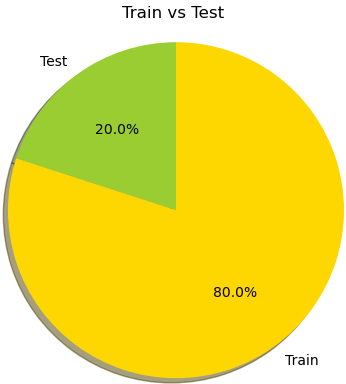
Pada tahapan ini tentunya peneliti membangun program dengan pengkodean pada tools jupyter notebook dari anaconda navigator dengan menggunakan algoritma Naïve Bayes, TF-IDF, dan CountVectorizer pada program pengkodean dan Bahasa pemrograman python dengan Bahasa pemrograman python versi 3.10.8.

**Hasil Pembahasan**

Hasil dari pada percobaan yang telah dilakukan dari masing-masing algoritma yang dapat dilihat pada table adalah sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| Pembagian Data (%) | | Akurasi (%) | |
| Data Training | Data Testing | | Naïve Bayes |
| 80% | 20% | | 95,65 |

Mengenai diagram terhadap akurasi ditas terhadap penggunaaan algoritma Naïve Bayes (NB) Classifier menghasilkan score akurasi sebesar 95,65% yang dimana diperoleh dari 80% data training dan 20% data testing.



Terlihat dari hasil percobaan kali ini. Dapat terlihat bahwa pembagian data dapat berpengaruh terhadap akurasi yang didapat dari setiap model. Namun pengruh pembagian data nya kecil sementara itu dengan pembagian data sebesar 80% data Training dan 20% data testing dapat diperoleh akurasi yang tinggi yaitu sebesar 95,65 dengan diperoleh algoritma naïve bayes, CountVectorizer, TF-IDF. Berdasarkan hasil dari analisis dan eksperimen yang telah dilakukan pada seluruh pengerjaan tugas Internship 1 ini dapat diperoleh hasil 95% dari tahap preprocessing hingga processing terhadap algoritma diatas

**DAGTAR PUSTAKA**

Aradea, S. A., Ariyan, Z., & Yuliana, A. (2011). Penerapan Decision Tree untuk Penentuan Pola Data Penerimaan Mahasiswa Baru. Jurnal Penelitian Sitrotika, 7(1).

Arifin, D. D., & Bijaksana, M. A. (2016). Enhancing spam detection on mobile phone Short Message Service (SMS) performance using FP-growth and Naive Bayes Classifier. 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), 80–84. IEEE.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 6(1), 20–29.

Bunker, R. P., & Thabtah, F. (2017). A machine learning framework for sport result prediction. Applied Computing and Informatics.

Dewi, I. N., & Supriyanto, C. (2013). Klasifikasi Teks Pesan Spam Menggunakan Algoritma Naïve Bayes. Semantik, 3(1).

Liu, B. (2007). Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media.

Padmanaban, H. (2014). Comparative analysis of Naive Bayes and tree augmented naïve Bayes models. Sethi, G., & Bhootna, V. (2014). SMS spam filtering application using Android. Int. J. Comput. Sci. Inf.Technol, 5(3), 4624–4626.

Srividhya, V., & Anitha, R. (2010). Evaluating preprocessing techniques in text categorization. International Journal of Computer Science and Application, 47(11), 49–51.

Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics, 1(1–4), 43–52.