IMPLEMENTASI *DATA MINING* UNTUK MEMPREDIKSI *CUSTOMER CHURN* MENGGUNAKAN ALGORITMA *NAIVE BAYES*

IMPLEMENTATION OF DATA MINING TO PREDICT CUSTOMER CHURNS USING NAIVE BAYES ALGORITHM

Risky Novendri[1], Rachmadita Andreswari[2], Oktariani Nurul Pratiwi[3]

[1,2,3] S1 Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom
[1]riskynovendri@student.telkomuniversity.ac.id , [2]andreswari@telkomuniveristy.co.id,
[3]onurulp@telkomuniversity.ac.id

Abstrak

Telkomsel merupakan perusahaan telekomunikasi yang paling banyak diminati oleh masyarakat Indonesia. Pada tahun 2018, perusahaan telkomsel memiliki jumlah pelanggan aktif sebanyak 163 juta pelanggan aktif. Namun, Tidak banyak pula pelanggan setia Telkomsel beralih ke operator lain. Dikabarkan bahwa, pada semester satu 2019 pelanggan telkomsel berkurang sebesar 5,7% dari yang awalnya 177,9 juta menjadi 167,8 juta pelanggan. Hal ini dikarenakan belum adanya suatu pemanfaatan data mining untuk memprediksi customer churn. Dengan memanfaatkan implementasi data mining menggunakan algoritma naive bayes untuk memprediksi customer churn. Sehingga, pada penelitian ini akan menggunakan algoritma naive bayes dan data total konsumsi kuota pelanggan setiap harinya selama satu bulan untuk memprediksi customer churn dan non-churn. Dari penelitian ini, peneliti mendapatkan hasil akurasi tertinggi sebesar 83,02%. Dari hasil prediksi pelanggan non-churn tersebut didapat hasil precision 84,90% dan recall 80,31% sehingga menghasilkan F1-measure sebesar 82,54%. Kemudian dari hasil prediksi customer churn t , diperoleh precision sebesar 81,43% dan recall 85,56 sehingga menghasilkan F1-measure sebesar 83,44%. Selain f1-measure, pada penelitian ini, menerapkan k-fold cross validation dan menghasilkan skor sebesar 82,94%. Dari hasil penelitian ini diharap dapat memberikan informasi yang bermanfaat bagi stakeholder terutama pihak perusahaan dalam pengambilan keputusan untuk mencegah terjadinya customer churn.
Kata kunci : Telkomsel, data mining, customer churn, *naive bayes, prediksi*.

Abstract

Telkomsel is a telecommunication company that is most in demand by Indonesians. In 2018, the Telkomsel company had a total of 163 million active subscribers. However, many Telkomsel loyal customers have switched to other operators. It is reported that in the first semester of 2019, Telkomsel subscribers decreased by 5.7% from 177.9 million to 167.8 million subscribers. This is due the lack of a data mining utilization to predict customer churn. By utilizing data mining implementation using naive bayes algorithm to predict customer churn. So, in this study will use naive bayes algorithm and data on total customer quota consumption every day for one month to predict customer churn and non-churn. From this study, researchers got the highest accuracy result of 83.02%. From the prediction results of non-churn customers, the precision results are 84.90% and the recall is 80.31%, resulting in an F1-measure of 82.54%. Then from the prediction results of customer churn t, a precision of 81.43% and a recall of 85.56 is obtained, resulting in an F1-measure of 83.44%. In addition to f1-measure, this study applies k-fold cross validation and produces a score of 82.94%. From the results of this study, it is hoped that it can provide useful information for stakeholders, especially the company in making decisions to prevent customer churn.

Keywords : Telkomsel, data mining, customer churn, *naive bayes, prediction*.

1. Pendahuluan

PT. Telekomunikasi Seluler (Telkomsel) merupakan salah satu perusahaan yang bergerak di bidang operator seluler pertama di Indonesia pada tahun 1995.Berdasarkan Laporan Tahunan Telkomsel, pada tahun 2018 Telkomsel memiliki 163 Juta pelanggan yang aktif [1]. Hingga saat ini Telkomsel merupakan pilihan operator yang sangat diminati oleh seluruh masyarakat Indonesia. Hal ini dikarenakan oleh banyaknya masyarakat percaya bahwa layanan seluler Telkomsel merupakan operator yang terbaik hingga saat ini.

Namun, walaupun Telkomsel merupakan operator yang sangat banyak diminati hingga saat ini. Tidak banyak pula pelanggan setia Telkomsel beralih ke operator lain. Menurut laporan dari CNBC Indonesia, Total pelanggan Telkomsel melorot 5,7% dari yang awalnya 177,9 juta menjadi 167,8 juta pada semester 1-2019[2]. Ada banyak alasan yang membuat *customer churn* menjadi masalah terbesar yang dihadapi oleh perusahaan. Salah satunya yaitu, mahalnya untuk memperoleh pelanggan baru tentunya membuat perusahaan lebih memilih mempertahankan pelanggan[3]. Kebocoran pendapatan terbesar di industri telekomunikasi dikarenakan meningkatnya perilaku pelanggan *churn* [4]. Banyak perusahaan menemukan alasan kehilangan pelanggan, dengan mengukur tingkat *customer churn* dan mendapatkan kembali pelanggan menjadi konsep yang sangat penting untuk mencegah customer *churn* (Herawati, 2016)

Dalam mempertahankan *customer*, perusahaan telekomunikasi membutuhkan cara untuk memprediksi untuk mengetahui risiko kapan *customer* akan menjadi *churn* (Hanifa, 2017). Peramalan *customer churn* dapat dilakukan dengan teknik Data *Mining*. *Data mining* merupakan suatu teknik mengumpulkan data untuk menemukan pola-pola tertentu. Dan untuk melakukan eksekusi Data *mining* dibutuhkan sebuah algoritma yang mampu untuk mengklasifikasikan *customer churn* atau non-churn. Untuk itu dapat disarankan untuk menggunakan algoritma *naive bayes*. Algoritma *naive bayes* merupakan algoritma yang sangat berguna untuk melakukan prediksi dengan menggunakan data yang sudah ada sebelumnya[7].

Naive bayes merupakan sebuah model klasifikasi yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas [7]. Algoritma *naive bayes* sering digunakan oleh para peneliti untuk melakukan prediksi terhadap suatu probabilitas. Maka, bukan tidak mungkin algoritma ini untuk melakukan prediksi *customer churn* karena algoritma *naive bayes* menghasilkan model probabilistik dari data yang diamati [8].

Berdasarkan dari permasalahan di atas dapat disimpulkan bahwa perlu adanya suatu prediksi terhadap pengguna yang akan beralih atau berhenti berlangganan (*Customer churn*) layanan operator jaringan Telkomsel. Oleh karena itu, pada penelitian ini akan membahas implementasi data mining untuk melakukan prediksi *customer churn* menggunakan algoritma *naive bayes.* Setelah berhasil melakukan prediksi *customer churn*, diharapkan dapat memberikan informasi yang bermanfaat bagi perusahaan dalam pengambilan keputusan.

2. Dasar Teori /Material dan Metodologi/perancangan

2.1. Customer Churn

Customer churn merupakan suatu keadaan yang cenderung untuk memberhentikan langganan dari produk atau jasa tertentu dalam suatu perusahaan [9]. *Customer churn* terjadi saat pelanggan atau subscriber berhenti berbisnis dengan perusahaan atau layanan[10]. Menurut Abdilla ada tiga jenis *Churn* [11]yaitu:

- Aktif : Pelanggan beralih ke produk atau layanan lain dikarenakan ketidakpuasan kualitas layanan.
- Rotasi / Insidental: Pelanggan berhenti menggunakan layanan namun tidak mengganti atau beralih layanan yang disebabkan oleh keadaan pelanggan tidak membutuhkan layanan tersebut.
- Discontinues : Bersifat pasif atau non-sukarela terhadap kontrak itu sendiri

2.2 Data Mining

Data *mining* adalah suatu proses yang digunakan untuk menemukan suatu informasi tersembunyi melalui pola atau tren yang ada pada basis data sehingga menemukan informasi baru yang sangat berguna[12], [13]. Data *mining* merupakan langkah penting dalam melakukan Knowledge Discovery from Data [12]. Terdapat beberapa tahapan proses yang harus dilalui dalam melakukan KDD menurut Han[13] di antaranya :

- Data *Cleansing* : Tahapan menghapus data yang tidak relevan
- Data Integration : Menggabungkan Banyak Data dari sumber berbeda
- Data Selection : Data yang relevan dianalisis
- Data Tranformation : Di mana data ditransformasikan dan dikonsolidasikan ke dalam bentuk yang sesuai untuk penambangan dengan melakukan operasi ringkasan atau agregasi
- Data *Mining* : Tahapan di mana proses untuk mengekstrak pola tertentu pada data
- Pattern Evaluation : Melakukan identifikasi terhadap pola-pola yang menarik yang dapat mewakili suatu pengetahuan
- Knowledge Presentation : Tahap terakhir di mana hasil pengetahuan akan di visualisasikan dan mempresentasikan pengetahuan untuk menyajikan hasil yang telah di capai

Selain Fungsi, Data *mining* dapat dilakukan dengan berbagai macam teknik yang berbeda berdasarkan dari hasil *mining*[14]. Data *mining* memiliki 4 jenis teknik yang dapat dilakukan untuk mengolah data untuk di pelajari yaitu [15]:

- *Supervised Learning* : Pada teknik ini dibutuhkan *training dataset*, Di mana variabel *input* akan menghasilkan variabel *output* sebagai target.
- Unsupervised Learning : Teknik ini hanya membutuhkan *dataset* variabel *input* dan tidak memiliki Variabel *output* sebagai target.
- Semisupervised Learning : Teknik Ini merupakan gabungan antara Supervise dan Unsupervised Learning. Pada teknik ini Memiliki variabel *input* yang lebih banyak namun memiliki variabel *output* target yang sedikit.

Kernel Learning : Merupakan teknik yang hanya dilakukan untuk mengekstrak, mewakili dan menganalisis suatu bentuk pola untuk mendukung mesin vektor.

2.3 Algoritma Naive Bayes

Algoritma *naive bayes* adalah pengklasifikasian statistik berdasarkan teorema Bayes yang dapat memprediksi probabilitas keanggotaan kelas seperti tupel tertentu yang memiliki kelas tertentu(Han,2012). Hristea[16] mengatakan bahwa *naive bayes* banyak digunakan karena efisiensi dan kemampuannya untuk menggabungkan bukti dari sejumlah besar fitur. Algoritma *naive bayes* memiliki keunggulan di mana data yang diperlukan hanya sedikit, namun dapat menghasilkan akurasi data yang tinggi dan pemrosesan data yang cepat[17]. Untuk menghitung setiap kelas keputusan pada *naive bayes* harus memiliki syarat bahwa kelas keputusan adalah benar (Olson, 2008). Algoritma *naive bayes* merupakan metode yang mengandalkan probabilitas. Sehingga pada teoremanya , Thomas Bayes mengemukakan teorinya berdasarkan pada rumus di bawah ini[8]:

$$\mathrm{P(A|B)} = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

Pada rumus di atas, B merupakan jumlah kelas yang belum diketahui kemudian A adalah kelas spesifik,P(A|B) adalah Probabilitas B berdasarkan kelas A,P(A)merupakan prior probabilitas A,P(B|A) probabilitas B berdasarkan A, dan yang terakhir yaitu P(B) yang merupakan probabilitas dari B. Penggunaan Algoritma ini, terdapat keunggulan di mana tidak memerlukan data yang banyak untuk melakukan *training* data[17]. Kemudian, Dikarenakan label yang pada penelitian ini berbentuk *binary*. Untuk persamaannya dapat di lihat sebagai berikut[19].

$$P(A|B) = \frac{1}{\sqrt{2\pi\sigma_A}} \exp\left(\frac{-(f_i-\mu_B)^2}{2\sigma_B^2}\right) \quad (2)$$

Dari rumus di atas, σ_B dan μ_B adalah estimasi yang menggunakan kemungkinan maksimum[19].

2.4 RFM Analisis

Analisis RFM adalah pendekatan klasik untuk memberikan penilaian dan mengelompokkan pelanggan ke dalam suatu segmentasi[20]. Menggunakan model RFM adalah cara untuk mengetahui pola interaksi dan perilaku pelanggan secara umum[21]. Dengan cara menghubungkan nilai R, F, M sebagai variabel independen dan dikelompokkan ke dalam suatu kategori[22].Pada RFM model memiliki tiga nilai yang harus diperhatikan[23] yaitu:

a. *Recency*(R) merupakan skor yang menilai transaksi terakhir yang pernah dilakukan oleh pelanggan

b. *Frequency*(F) merupakan penilaian seberapa banyak pelanggan melakukan transaksi dalam kurun waktu tertentu.

c. *Monetary Value*(M) yaitu menghitung seberapa banyak pelanggan menghabiskan uangnya untuk berbelanja selama rentang waktu tertentu.

Model RFM menghasilkan skor yang memberi peringkat kepada pelanggan satu sama lain dan pelanggan yang memiliki skor paling tinggi merupakan pelanggan yang berharga[24].Untuk menghitung nilai skor RFM adalah dengan menggunakan teknik nested *binning* atau independen *binning* di mana nilai yang bervariasi akan dikelompokkan ke dalam rentang tertentu[25]. Kemudian, dari nilai RFM yang sudah di *binning* tersebut akan di jumlah kan seluruhnya sehingga dapat mengelompokkan customer berdasarkan skor yang di peroleh.

2.5 Confusion Matrix

Confusion matrix adalah *matrix* yang berfungsi sebagai penentuan kualitas model machine learning yang direpresentasikan melalui baris dan kolom, di mana baris mewakili nilai aktual sedangkan kolom mewakili kelas prediksi[26].

Tabel 1 Konsep Confusion Matrix

Klasifikasi	***True***	***False***
True	*True* Positif	*False* Positif
False	*False* Negatif	*True* Negatif

Menurut Santosa[8], Apabila nilai *True* yang diklasifikasikan benar maka disebut sebagai *True* Positif(TP) dan Apabila Kelas Negatif diklasifikasikan benar maka disebut sebagai *True* Negatif(TN). Sebaliknya, Apabila nilai positif yang diklasifikasikan salah maka disebut sebagai *False* Positif(FP) dan Apabila Kelas Negatif diklasifikasikan salah maka disebut sebagai *False* Negatif(FN).

2.6 F1-Measure

F1-*measure* adalah pertukaran antara mengklasifikasikan semua poin data dengan benar dan memastikan bahwa setiap kelas hanya berisi poin dari satu kelas [27]. F-*measure* juga merupakan suatu metode untuk mengevaluasi suatu model dan memtukan kualitas suatu model [28]. Untuk menghitung F-*measure* memerlukan *recall* dan *precision* [29]. Untuk menghitung nilai f1-*measure* yaitu dengan menggunakan rumus [6]:

$$F - Measure = \frac{2(Precision * Recall)}{Precision + Recall} \quad (3)$$

Pada rumus di atas merupakan rata-rata harmonis antara *precision* dan *recall* yang menghasilkan nilai F1-*measure*. *Precision* adalah nilai yang menghitung nilai prediksi positif sedangkan *recall* menunjukkan bahwa probabilitas semua nilai yang relevan dipilih oleh model [27]. Pada *recall* akan menggunakan perhitungan sebagai berikut [12]:

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

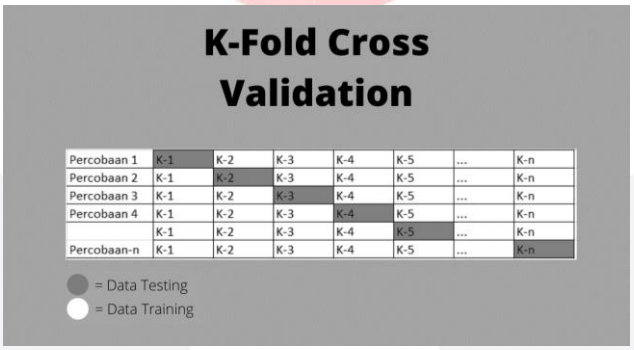
Sedangkan, untuk menghitung nilai *precision* adalah dengan rumus:

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Sehingga, setelah kita mendapatkan nilai *precision* dan *recall*, kita dapat menghitung nilai f1-*measure* untuk mengevaluasi model. Suatu model dapat dikatakan berkualitas sangat baik apabila nilai f1-*measure* mendekati 100% sebaliknya performa model dikatakan buruk apabila mendekati 0%[6].

2.7 K-fold Cross Validation

K-fold cross validation merupakan suatu metode pendekatan statistik yang memverifikasi kinerja prediksi pada suatu model di mana semua sampel akan dikelompokkan ke dalam kelompok k, di mana k-1 akan menjadi data latih dan sisanya untuk menguji keakuratan model [30]. Pada *k-fold* cv, pelatihan akan di lakukan sebanyak k *subset* di mana masing-masing data asli akan di bagi dengan ukuran yang sama [12], [13]. Dalam penerapan k-*fold*, umumnya dan direkomendasikan menggunakan sebanyak 10 *k-fold*[12], [13], [31]. Untuk ilustrasi dari penjabaran pada penjelasan sebelumnya, dapat di lihat pada gambar di bawah ini.

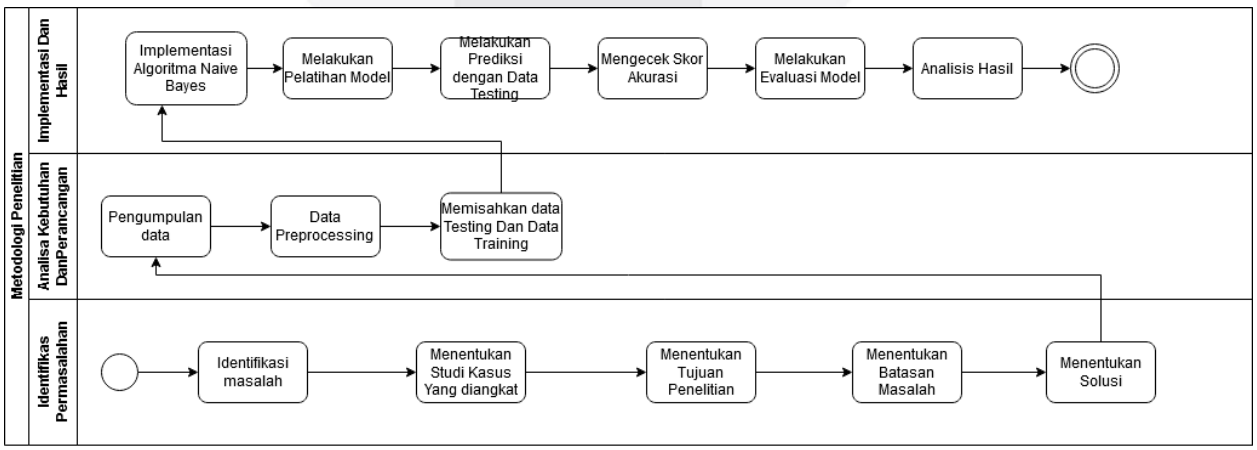


Gambar 1 Konsep K-fold Cross Validation

Dari gambar di atas, dapat di lihat bahwa, **k-n** merupakan jumlah atau banyaknya *fold* yang ingin kita tentukan. Sehingga apa bila kita menggunakan **k-n**, maka percobaan akan dilakukan sebanyak **n** kali. Dan data yang akan divalidasi akan dibagi sebanyak **n**. Kemudian, dari banyaknya percobaan **n** masing-masing **k-n** akan menjadi data testing dan sisanya akan menjadi data training.

3. Metodologi

Dalam pengerjaan penelitian ini terdiri dari tiga tahapan diantaranya yaitu Identifikasi Masa.



3.1 Dataset

Data yang digunakan peneliti diperoleh dari perusahaan yang memiliki akses database Telkomsel. Data tersebut merupakan data log customer dari tanggal 19 juni 2020 hingga 19 juli 2020. Dikarenakan data yang akan digunakan

adalah berformat zip maka peneliti akan melakukan ekstraksi *dataset*. Pada saat ektraksi *dataset*, data yang dihasilkan setelah proses ektraksi adalah berformat txt. Kemudian, data tersebut akan di tampilkan ke dalam objek *dataframe*. Dari total 109 kolom yang ada, peneliti hanya akan menggunakan 3 kolom dari data *basic* yang dapat di lihat pada tabel di bawah ini.

Kolom *Dataset*	Deskripsi
MSIDN	MSIDN adalah Mobile Subscriber Integrated Services Digital Network Number yang merupakan nomor ponsel customer telkomsel
IMSI	IMSI adalah International Mobile Subscriber Identity yang berfungsi sebagai identifikasi pengguna pada jaringan
Total Consumption	Total consumption merupakan kolom yang berisikan jumlah konsumsi customer per harinya

Dikarenakan data yang diberikan disensor, maka peneliti akan menggunakan kolom MSIDN dan IMSI sebagai ID. Selain itu, Kolom Total konsumsi akan digunakan untuk melihat pola konsumsi data yang dimiliki oleh customer. Dikarenakan penelitian ini untuk memprediksi pola customer selama 1 bulan, maka peneliti akan menggabungkan seluruh *dataset* yang sudah di ekstrak dan di seleksi pada tahap sebelumnya, Kemudian dari data *basic* yang awalnya 31 berkas akan di gabungkan menjadi satu berkas csv. Setelah data digabungkan menjadi satu, peneliti dapat mengamati keseluruhan total konsumsi *customer* sehingga *dataset* menjadi seperti di bawah ini:

MSIDN	**IMSI**	**Tc_day1**	**Tc_day-n**	**....**	**Tc_day31**
6281318059XXX	510101832059XXX	202.0	2164303.0	...	1075032.0
X2812X10X80X	51010X1250X80XX	137416.0	22077904.0	...	3906377.0
6281192209XX	X1010921321X0XX	2940787.0	3890527.0	...	0.0
628XX8XX678X	5X0X08XX2X678X	388641.0	51652.0	...	31973.0
6XXX1X35737X	51010X36X5737XX	18034090.0	2024309.0	...	2836020.0

Data kolom Tc_day1 hingga Tc_day-31 yang digunakan adalah untuk mengetahui pola yang dimiliki oleh *customer* non-churn dan *customer churn* selama satu bulan. Kemudian, peneliti melakukan export berkas yang sudah digabungkan ke dalam bentuk format berkas csv.

3.2 Data Pre-processing

Pada tahap ini data akan melalui beberapa tahap agar data dapat diterima atau digunakan oleh model.

3.2.1 RFM Analisis

Setelah menggabungkan data pada tahap sebelumnya, selanjutnya adalah melakukan RFM analisis untuk menentukan label *customer churn* atau Non-churn. Untuk menentukan label, dapat menggunakan RFM *Score* sebagai acuan untuk klasifikasi *customer churn* atau non-churn hal ini sama seperti yang dilakukan oleh peneliti sebelumnya[20]. Untuk dapat menghitung skor maka diperlukannya kolom R(*recencies*),F(*frequency*) dan M(*monetary value*). Karena itu, tahap ini akan menghitung nilai RFM pada masing-masing kolom seperti yang dilakukan oleh beberapa peneliti[20], [24], [32].

Tabel 2 Nilai RFM

R	**F**	**M**
31	21	402547.509000
31	22	158951.592750
30	20	101280.105500
30	16	34911.497375
31	19	57436.515750

untuk mendapatkan nilai R adalah dengan mengecek kapan terakhir kali *customer* konsumsi data dari 31 hari terakhir. Kemudian, untuk mendapatkan nilai F adalah dengan menghitung berapa banyak pengguna aktif dalam satu bulan. Dan terakhir adalah nilai M yang didapat dengan menjumlahkan total konsumsi selama satu bulan.

Dikarenakan banyaknya variasi nilai kolom RFM, dapat menyebabkan sulitnya untuk menghitung RFM Score. Untuk mengatasi hal tersebut, dapat diatasi dengan melakukan *binning* untuk menentukan rentang dari masing-masing kolom RFM agar dapat mengelompokan skor masing-masing kolom ke dalam rentang tertentu [25]. Pada tahap ini, Skor akan dibagi ke dalam rentang 1 sampai 10, di mana 1 adalah nilai yang paling rendah dan 10 adalah nilai yang paling tinggi. Setelah mendapatkan nilai bining RFM, tahap selanjutnya adalah melakukan konversi tipe data yang ada pada kolom *binning* tersebut dikarenakan data yang dihasilkan adalah tipe data kategori. Tujuan untuk melakukan konversi tipe data adalah agar kolom tersebut dapat dijumlahkan sehingga menghasilkan kolom RFM *score*[20], [23]. Kemudian dari hasil konversi tipe data, untuk menentukan skor dari RFM adalah dengan menghitung jumlah total skor yang didapat dari kolom RFM *binning*. Sehingga data akan terlihat seperti pada tabel di bawah ini.

Tabel 3 Binning RFM

R	**F**	**M**	**r_bin**	**f_bin**	**m_bin**	***Score***
31	21	402547.509000	10.0	7.0	4.0	21.0
31	22	158951.592750	10.0	7.0	2.0	19.0
30	20	101280.105500	10.0	6.0	1.0	17.0
30	16	34911.497375	10.0	5.0	1.0	16.0
31	19	57436.515750	10.0	6.0	1.0	17.0

Setelah data menghasilkan rfm *score*, selanjutnya data akan masuk ke dalam tahap pre-processing, yaitu mendapatkan kolom label yang berisikan nilai *customer churn* atau non-churn. Sehingga kolom Label merupakan hasil *binning* dari nilai kolom RFM *score* yang dihasilkan sebelumnya. Untuk lebih rinci dapat di lihat penjelasan pada tabel di bawah ini.

Tabel 4 Deskripsi Label

Nilai	Deskripsi
Non-churn	Non-churn merupakan label yang diberikan kepada pelanggan yang memiliki *Score* RFM > 16
Churn	Merupakan Label yang diberikan oleh pelanggan yang memiliki *Score* RFM <= 16

Pada kolom label dapat di lihat bahwa terdiri dari 2 nilai yaitu non-churn yang memiliki nilai RFM > 16 Sedangkan nilai yang memiliki <= 16 akan masuk ke dalam kategori *churn*. Dikarenakan score yang terendah merupakan nilai segmentasi yang paling buruk[24], [33]. Maka dapat diasumsikan bahwa nilai segmentasi terendah memiliki kemungkinan besar berpotensi untuk churn terhadap terhadap perusahaan. Oleh karena itu, untuk membedakan customer churn dan non-churn adalah dengan menggunakan nilai kuartil satu (Q1) sebagai acuan untuk mengklasifikasikan customer churn. Dari keseluruhan nilai RFM Score, Skor 16 merupakan nilai kuartil 1 (Q1) atau batas terendah dari keseluruhan nilai RFM Score. Sehingga hasil dari RFM analisis adalah kolom label yang akan digunakan untuk melakukan prediksi *customer churn*.

3.2.2 Data Cleansing

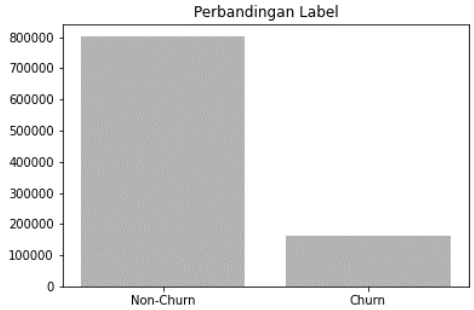
Data *cleansing* merupakan tahap mengisi, menghaluskan dan mengoreksi data yang tidak konsisten dalam data [13]. Pada tahap ini, *dataset* pada penelitian ini memiliki nilai Nan atau *null* dan terdapatnya nilai yang tidak seimbang terhadap nilai label. Sebelum itu, pada dataset akan dilakukan drop kolom RFM dikarenakan pada analisis RFM hanya akan mengambil kolom label untuk klasifikasi *customer*.

a. Mengatasi Missing value

Nan atau *null* merupakan istilah untuk data yang hilang atau tidak berhasil didapat pada saat pengambilan data sehingga dapat mengurangi kualitas data yang dimiliki. Sehingga, data *cleansing* bertujuan agar *dataset* yang dimiliki tidak mengurangi kualitas data. Dan apabila di drop maka akan membuat jumlah *dataset* yang dimiliki akan menurun drastis. Hal ini dikemukakan oleh Han(2012), Yaitu dengan menggunakan konstanta global untuk mengisi nilai yang hilang, di mana semua atribut yang memiliki *missing value* akan diisi dengan unknown,0 atau ∞.

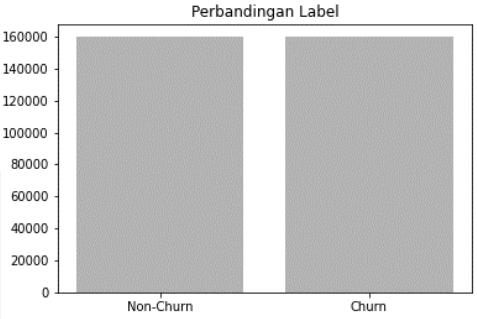
b. Mengatasi Data Imbalance

Pada *dataset* penelitian ini, memiliki nilai yang dominan terhadap nilai non-churn sehingga nilai *churn* memiliki mayoritas lebih sedikit dari total jumlah data.Untuk lebih jelasnya dapat melihat gambar di bawah ini.



Gambar 2 Perbandingan Nilai Kolom Label

Dikarenakan data dengan nilai non-churn lebih banyak daripada data bernilai *churn* maka, dapat menerapkan metode *undersampling* untuk mengatasi data yang tidak seimbang tersebut. *Undersampling* merupakan suatu metode untuk mengatasi data yang tidak seimbang dengan mengurangi nilai mayoritas sehingga jumlahnya sama dengan nilai yang minoritas Han(2012). Hal ini juga dilakukan oleh peneliti sebelumnya untuk mengatasi data yang tidak seimbang(Hanifa,2017). Sehingga untuk mengatasi data yang tidak seimbang dapat melakukan drop pada data yang bernilai non-churn agar memiliki jumlah total yang sama dengan nilai *churn*. Setelah menerapkan undersampling,



Gambar 3 Hasil Penanganan Imbalance Data

selanjutnya adalah menggabungkan data tersebut menjadi objek *dataframe* yang baru. Sehingga data yang sebelumnya tidak seimbang setelah memasuki tahap undersampling data menjadi seimbang. Untuk hasil tahap undersampling dapat di lihat pada gambar di bawah ini.

Dapat di lihat pada gambar di atas, data yang awalnya tidak seimbang telah di atasi dengan metode undersampling sehingga data menjadi seimbang. Hal ini dilakukan agar dapat meningkatkan performa pada saat *training* model.

3.3 Pembagian Dataset

Pada tahap ini, *dataset* akan dibagi menjadi dua bagian. Dua bagian itu terdiri dari data *training* dan data *testing* [6], [8]. Dari keseluruhan *dataset* akan dibagi menjadi beberapa rasio untuk menghitung akurasi terbaik untuk data *testing* dari total keseluruhan 320000 baris data. Untuk lebih jelasnya dapat di lihat pada tabel di bawah ini.

Tabel 5 Rasio Perbandingan Data

Rasio	**Data *Training***	**Data *Testing***	**Total**
7:3	224000	96000	320000
8:2	256000	64000	320000
5:5	160000	160000	320000

Dapat di lihat pada tabel di atas, *dataset* terdiri dari 3 rasio perbandingan di antaranya adalah rasio 7:3 di mana data *training* memiliki jumlah 224000 baris data dan data *testing* 96000 baris data. Kemudian perbandingan 8:2 yang memiliki data *training* 256000 dan 64000 baris data *testing*. Dan yang terakhir adalah rasio 5:5 yang memiliki data *training* dan data *testing* masing-masing 160000 baris data. Dari data ini nantinya akan membandingkan hasil akurasi dari masing-masing rasio data. Setelah data disesuaikan terhadap model, maka data akan siap masuk ke tahap implementasi.

3.4 Implementasi Naive Bayes

Model yang digunakan pada penelitian ini akan menggunakan fungsi make_pipeline pada library sklearn untuk melakukan untuk mengumpulkan langkah yang dapat divalidasi silang dengan parameter yang berbeda secara

otomatis. Hal ini sama seperti yang dilakukan oleh peneliti sebelumnya dalam menerapkan model[34]. Sehingga di dalam fungsi make_pipeline terdiri dari quantile transformer dan gaussian *naive bayes*. Quantile tranformer adalah

```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import QuantileTransformer
pipeline = make_pipeline(QuantileTransformer(output_distribution='uniform'), GaussianNB())
model = pipeline.fit(xtrain,ytrain)
model
```

sebuah fungsi tranformasi kuantitatif yang berfungsi untuk memetakan distribusi probabilitas variabel ke distribusi probabilitas lain. Sehingga probabilitas variabel input atau atribut terdistribusi dengan baik dan dapat meningkatkan kualitas model. Untuk lebih jelasnya dapat melihat kode berikut:

Pada kode di atas, maka secara otomatis model akan melakukan *training* terhadap data yang sudah diolah sebelumnya. Setelah melakukan tahap *training*, model yang sudah di *training* akan melakukan prediksi. Kemudian, data hasil prediksi akan di cek akurasinya.

3.5 Hasil Akurasi

Seperti yang dijelaskan sebelumnya, tahap ini akan menguji kualitas akurasi terhadap pembagian rasio data *training* dan data *testing* untuk mengecek kualitas terbaik yang dihasil kan oleh model. Untuk mengecek hasil akurasi, pertama data *testing* akan di coba untuk melakukan prediksi. Setelah hasil prediksi didapat, selanjutnya yaitu melakukan validasi data menggunakan fungsi accuracy_*score* pada label data *testing*[7], [8], [34], [35].

Dataset dalam pemodelan ini akan dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Pembagian data *training* dan data *testing* dengan perbandingan rasio 7:3, 8:2 dan 5:5 dari total 320000 baris data. Kemudian, dari data tersebut akan diuji untuk melihat hasil akurasi yang terbaik. Untuk hasil *testing* yang dapat diilihat pada tabel di bawah ini:

Tabel 6 Hasil Akurasi

Rasio	Data *Training*	Data *Testing*	Akurasi	Jumlah Data
7:3	224000	96000	83.02%	320000
8:2	256000	64000	82.91%	320000
5:5	160000	160000	82,98%	320000

Pada tabel di atas dapat diihat bahwa hasil paling baik adalah dengan rasio 7:3 yang menghasilkan akurasi sebesar 83.02%. Kemudian, rasio 5:5 menghasilkan 82,98% dan rasio 8:2 menghasilkan akurasi sebesar 82,91%. Dari hasil pengujian akurasi tersebut, penelitian ini akan mengevaluasi hasil akurasi tertinggi yaitu dengan rasio 7:3.

3.6 Evaluasi Model

Dari hasil pengujian akurasi, selanjutnya model akan dievaluasi hasil prediksinya menggunakan *confusion matrix*[8]. Setelah diterapkan ke dalam *confusion matrix* maka akan menghasilkan data yang dapat di lihat dari tabel di bawah ini:

Tabel 7 Confusion Matrix Pada Model

	Hasil Prediksi	
	Non-churn	***Churn***
Non-churn	38551	6853
Churn	9449	41147

Berdasarkan *confusion matrix* yang didapat, dapat disimpulkan bahwa :

a. Hasil prediksi terhadap pelanggan non-churn yang benar (*True*-Positif) adalah sebanyak 38551 prediksi. Sedangkan, hasil prediksi yang salah(*False*- Positif) memiliki jumlah sebanyak 6853

b. Hasil prediksi terhadap pelanggan *churn* yang benar (*True*-Negatif) adalah sebanyak 41147 prediksi. Sedangkan, hasil prediksi yang salah(*False*- Negatif) memiliki jumlah sebanyak 9449

Dari hasil di atas, kita dapat menghitung akurasi yang didapat yaitu dengan menghitung dengan perhitungan yang dijelaskan pada bab sebelumnya sebagai berikut:

$$Accuracy = \frac{38551 + 41147}{38551 + 41147 + 9449 + 6852}$$

$$Accuracy = \frac{79.698}{960000}$$

$$Accuracy = 0{,}8302$$

Dari hasil perhitungan di atas, akurasi yang didapat oleh model adalah sebesar 0,8302 atau 83.02%.

3.6.1 F1-measure

Setelah mendapatkan hasil *confusion matrix*, Selanjutnya adalah menghitung nilai *recall*, *precision* dan f1-*score*[6]. *Precision* berfungsi untuk mengetahui jumlah perbandingan data yang berhasil di prediksi dari keseluruhan data prediksi. Untuk perhitungan nilai *precision* pelanggan non-churn yang dapat di lihat di bawah ini.

$$Precision\ True = \frac{38551}{38551 + 6852}$$

$$Precision\ True = \frac{38551}{45403}$$

$$Precision\ True = 0{,}8490 = 84{,}90\%$$

Dari perhitungan di atas, nilai *precision* yang dihasilkan adalah sebesar 84,90% Kemudian, *recall* adalah untuk mengetahui perbandingan antara jumlah hasil prediksi dari keseluruhan data sebenarnya. Untuk menghitungnya dapat di lihat pada perhitungan berikut.

$$Recall\ True = \frac{38551}{38551 + 9449}$$

$$Recall\ True = \frac{38551}{48000}$$

$$Recall\ True = 0{,}8031$$

Dari perhitungan *recall* yang dihasilkan terhadap model adalah sebesar 80,31%. Setelah mendapatkan nilai *precision* dan *recall*, maka selanjutnya adalah menghitung nilai f1-*score*. Untuk perhitungannya dapat di lihat di bawah ini.

$$F1 - Score\ True = \frac{2(0{,}8490 * 0{,}8031)}{0{,}8490 + 0{,}8031}$$

$$F1 - Score\ True = \frac{1{,}3636638}{1{,}6521}$$

$$F1 - Score\ True = 0{,}8254$$

Dari perhitungan F1-s*core* terhadap nilai *true* menghasilkan nilai f1-*score* sebesar 82,54%. Selain menghitung nilai *precision*, *recall* dan f1- *score true*, tahap ini juga menghitung nilai yang sama terhadap nilai *false*. Untuk Hasil lebih lengkapnya dapat di lihat pada tabel di bawah ini:

Tabel 8 F1-Score Pada Label

Label	***Precision***	***Recall***	**F1-*Score***
Non-churn	84,90%	80,31%	82,54%
Churn	81,43%	85,56%	83,44%

Dari hasil tersebut dapat dijelaskan bahwa model berhasil dengan tepat memprediksi label *non-churn* sebesar 84,90% (*precision*) dari yang seharusnya diprediksi *non-churn* dan juga kemampuan model dalam menemukan label yang relevan dengan *non-churn* adalah sebesar 80,39% (*recall*). Sehingga pada model menghasilkan keseimbangan antara *precision* dan *recall* sebesar 82,49% (*f1-score*). Selain itu, dari hasil tersebut dapat dijelaskan bahwa model berhasil dengan tepat memprediksi label *churn* sebesar 81,43% (*precision*) dari yang seharusnya diprediksi *churn* dan juga kemampuan model dalam menemukan label yang relevan dengan *churn* adalah sebesar 85,56% (*recall*). Sehingga pada model menghasilkan keseimbangan antara *precision* dan *recall* sebesar 83,44% (*f1-score*). Dikarenakan keseluruhan skor memiliki hasil besar dari 80% berdasarkan skor yang diperoleh, maka hasil evaluasi masuk ke dalam kategori baik.

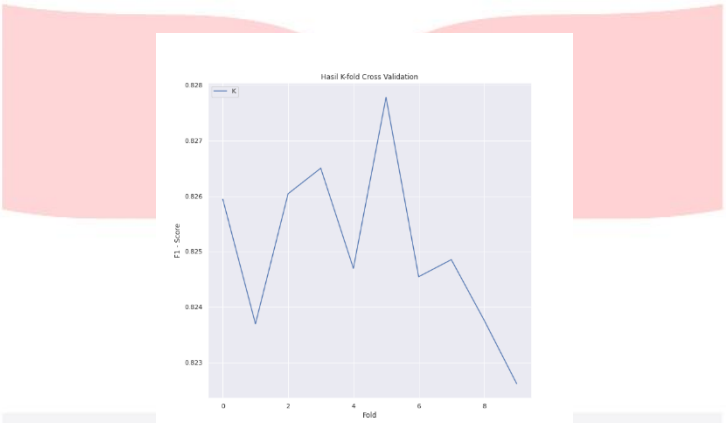
3.6.2 K-fold Cross Validation

Selain menggunakan f1-*measure*, peneliti akan menggunakan *k-fold cross validation* untuk memverifikasi kinerja model dalam melakukan prediksi. Dalam penerapannya, peneliti akan menggunakan *k-fold* sebanyak 10 seperti yang

disarankan oleh peneliti sebelumnya (Han, 2012; Jung, 2018; Larose, 2014). Untuk menerapkannya, peneliti menjalan kode sebagai berikut.

```
cv = KFold(n_splits=10, random_state=1, shuffle=
True)
Score = cross_val_score(model, x, y, cv=cv, scor
ing='f1').mean()
print('Kfold Cross validated Score : {:.4f}'.for
```

Code di atas merupakan kode untuk penggunaan *k-fold cross validation* dan pada *code* tersebut akan menghitung nilai *f1-score* pada model yang dihasilkan sebelumnya. Hal ini dikarenakan sebelumnya peneliti menghitung nilai *f1-score* pada model agar sejalan dengan evaluasi sebelumnya sehingga, peneliti menetapkan *scoring* dengan *f1-score* pada *k-fold cross validation*. Setelah menjalankan kode di atas, peneliti mendapatkan hasil skor yang dapat dilihat pada gambar berikut.



Gambar 4 Hasil K-fold Cross Validation

Dari gambar di atas, merupakan hasil validasi silang dari model dengan menggunakan 10 *k-fold*. Dari penerapan *k-fold cross validation* tersebut diperoleh rata-rata skor sebesar 82,50%. Berdasarkan persentase skor yang diperoleh, maka skor *k-fold cross validation* masuk ke dalam klasifikasi baik.

4. Kesimpulan Dan Saran

4.1 Kesimpulan

Berdasarkan dari hasil penelitian ini dapat disimpulkan bahwa implementasi data *mining* untuk memprediksi *customer churn* menggunakan algoritma *naive bayes*, dapat dilakukan dengan sangat baik. Dari hasil pengujian akurasi, hasil prediksi yang dihasilkan tidak terlalu berpengaruh terhadap perbedaan rasio data yang digunakan. Sehingga berapa pun rasio data yang dipakai, tidak akan berpengaruh terhadap model yang akan dilatih menggunakan algoritma *naive bayes*. Hal ini dikarenakan, algoritma *naive bayes* merupakan algoritma yang menggunakan prediksi dengan perhitungan probabilitas.

Kemudian, algoritma *naive bayes* menghasilkan akurasi tertinggi 83,02%. Di mana hasil dari akurasi tersebut memprediksi pelanggan non-churn yang benar sebesar 38551 prediksi hasil prediksi pelanggan non-churn yang salah memiliki jumlah sebanyak 9449 prediksi. Dari hasil prediksi pelanggan non-churn tersebut didapat hasil *precision* 84,90% dan *recall* 80,31% sehingga menghasilkan F1-*measure* sebesar 82,54%. Kemudian untuk prediksi *customer churn*, model memprediksi pelanggan *churn* yang benar adalah 41147 prediksi dan hasil prediksi yang salah memiliki jumlah sebanyak 6853 prediksi. Kemudian dari hasil prediksi *customer churn* tersebut, diperoleh *precision* sebesar 81,43% dan *recall* 85,56 sehingga menghasilkan F1-*measure* sebesar 83,44%. Selain f1-*measure*, dan supaya hasil prediksi lebih valid, peneliti menggunakan k-*fold* cross *validation*. Pada penelitian ini, penerapan k-*fold* cross *validation* menghasilkan *score* sebear 82,94%.

4.2 Saran

Untuk memudahkan peneliti dalam melakukan RFM analisis, disarankan untuk menggunakan *dataset* yang memiliki riwayat transaksi yang memiliki kolom data transaksi terakhir, jumlah total pengeluaran dan juga data log *customer* lebih dari 6 bulan sehingga dapat dilakukan RFM analisis yang lebih baik. Kemudian, Untuk penelitian selanjutnya diharapkan dapat menyempurnakan penelitian ini dengan menggunakan algoritma lain seperti SVM, *Decision Tree*, Atau Algoritma klasifikasi yang lainnya. Dan apabila data yang *dataset* yang memiliki baris yang sangat banyak, lebih baik menggunakan algoritma *Artificial Neural Network* karena ANN lebih cocok digunakan terhadap *dataset* yang

besar. Sehingga peneliti dapat membandingkan algoritma yang lebih baik dalam melakukan prediksi *customer churn*. Selanjutnya peneliti berharap ke depannya ada suatu cara untuk meningkatkan performa model yang dimiliki oleh peneliti saat ini. Dan juga peneliti berharap hasil penelitian ini dapat digunakan sebaik-baiknya bagi seluruh peneliti maupun perusahaan untuk menerapkan prediksi *customer churn* sebagai pengambilan keputusan untuk mencegah *customer churn*.

Referensi

[1] Telkomsel, "Telkomsel Annual Report 2018: Your Gateway To The Digital World," 2019, [Online]. Available: https://www.telkomsel.com/about-us/investor-relations.

[2] S. Sidik, "Total Pelanggan Telkomsel Anjlok Jadi 167,7 Juta, Kenapa?," *www.cnbcindonesia.com*, 2019. https://www.cnbcindonesia.com/market/20190814232116-17-92092/total-pelanggan-telkomsel-anjlok-jadi-1677-juta-kenapa.

[3] M. Arifin, "Ig-Knn Untuk Prediksi Customer Churn Telekomunikasi," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 6, no. 1, p. 1, 2015, doi: 10.24176/simet.v6i1.230.

[4] R. . . Jadhav, "Churn Prediction in Telecommunication Using Data Mining Technology," *Int. J. Adv. Comput. Sci. Appl.*, vol. 2, no. 2, pp. 17–19, 2011, doi: 10.14569/ijacsa.2011.020204.

[5] M. Herawati, I. L. Wibowo, and I. Mukhlash, "Prediksi Customer Churn Menggunakan Algoritma Fuzzy Iterative Dichotomiser 3," *Limits J. Math. Its Appl.*, vol. 13, no. 1, p. 23, 2016, doi: 10.12962/j1829605x.v13i1.1913.

[6] T. T. Hanifa, "Analisis Churn Prediction pada Data Pelanggan PT . Telekomunikasi dengan Logistic Regression dan Underbagging," *Univ. Telkom*, vol. 4, no. 2, p. 78, 2017.

[7] Kaharudin, "PREDIKSI CUSTOMER CHURN PERUSAHAAN TELEKOMUNIKASI MENGGUNAKAN NAÏVE BAYES DAN K-NEAREST NEIGHBOR," *Magister Tek. Inform. Univ. AMIKOM Yogyakarta*, 2019.

[8] S. Santosa and R. Yuliantara, "Model Prediksi Pola Loyalitas Pelanggan Telekomunikasi Menggunakan Naive Bayes Dengan Optimasi Particle Swarm Optimization," *J. Teknol. Inf.*, vol. 13, pp. 154–169, 2017.

[9] D. H. Tisantri, "Prediksi Keputusan Pelanggan Menggunakan Extreme Learning Machine Pada Data Telco Customer Churn," vol. 3, no. 11, p. 8, 2019.

[10] M. Galetto, "What Is Customer Churn?," *www.ngdata.com*, 2016. https://www.ngdata.com/what-is-customer-churn/.

[11] M. F. Abdillah, "Penggunaan Deep Learning untuk Prediksi Churn pada Jaringan Telekomunikasi Mobile," vol. 3, no. 2, pp. 3882–3888, 2016.

[12] D. T. Larose, *Discovering Knowledge in Data*. 2014.

[13] J. Han, *Data mining: Data mining concepts and techniques*, 3rd ed. USA: Morgan Kaufmann, 2012.

[14] R. Suresh and S. R. Harshni, "Data mining and text mining - A survey," *6th Int. Conf. Comput. Power, Energy, Inf. Commun. ICCPEIC 2017*, vol. 2018-Janua, no. i, pp. 412–419, 2018, doi: 10.1109/ICCPEIC.2017.8290404.

[15] R. Sullivan, *Introduction to data mining for the life sciences*, vol. 9781597452. 2012.

[16] F. T. Hristea, *The Naïve Bayes Model for Unsupervised Word Sense Disambiguation*. New York: Springer, 2013.

[17] S. Chen, G. I. Webb, L. Liu, and X. Ma, "A novel selective naïve Bayes algorithm," *Knowledge-Based Syst.*, vol. 192, no. xxxx, p. 105361, 2020, doi: 10.1016/j.knosys.2019.105361.

[18] D. L. Olson and D. Delen, *Advanced data mining techniques [electronic resource]*. 2008.

[19] L. Ali *et al.*, "A Feature-Driven Decision Support System for Heart Failure Prediction Based on $\chi 2$ Statistical Model and Gaussian Naive Bayes," *Comput. Math. Methods Med.*, vol. 2019, 2019, doi: 10.1155/2019/6314328.

[20] Y. Aleksandrova, "Application of machine learning for churn prediction based on transactional data (RFM analysis)," *Int. Multidiscip. Sci. GeoConference Surv. Geol. Min. Ecol. Manag. SGEM*, vol. 18, no. 2.1, pp. 125–132, 2018, doi: 10.5593/sgem2018/2.1/S07.016.

[21] S. Mitrović, G. Singh, B. Baesens, W. Lemahieu, and J. De Weerdt, "Scalable RFM-Enriched representation learning for churn prediction," *Proc. - 2017 Int. Conf. Data Sci. Adv. Anal. DSAA 2017*, vol. 2018-Janua, pp. 79–88, 2017, doi: 10.1109/DSAA.2017.42.

[22] R. G. Martínez, R. A. Carrasco, J. García-Madariaga, C. P. Gallego, and E. Herrera-Viedma, "A comparison between Fuzzy Linguistic RFM Model and traditional RFM model applied to Campaign Management. Case study of retail business.," *Procedia Comput. Sci.*, vol. 162, no. Itqm, pp. 281–289, 2019, doi: 10.1016/j.procs.2019.11.286.

[23] A. Sheshasaayee and L. Logeshwari, "Implementation of Clustering Technique Based RFM Analysis for Customer Behaviour in Online Transactions," *Proc. 2nd Int. Conf. Trends Electron. Informatics, ICOEI 2018*, no. Icoei, pp. 1166–1170, 2018, doi: 10.1109/ICOEI.2018.8553873.

[24] İ. SABUNCU, E. TÜRKAN, and H. POLAT, "Customer Segmentation and Profiling With Rfm Analysis," *Turkish J. Mark.*, vol. 5, no. 1, pp. 22–36, 2020, doi: 10.30685/tujom.v5i1.84.

[25] M. Mohammadzadeh, Z. Z. Hoseini, and H. Derafshi, "A data mining approach for modeling churn behavior

via RFM model in specialized clinics Case study: A public sector hospital in Tehran," *Procedia Comput. Sci.*, vol. 120, pp. 23–30, 2017, doi: 10.1016/j.procs.2017.11.206.

[26] O. Caelen, "A Bayesian interpretation of the confusion matrix," *Ann. Math. Artif. Intell.*, vol. 81, no. 3–4, pp. 429–450, 2017, doi: 10.1007/s10472-017-9564-8.

[27] I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam, and S. W. Kim, "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector," *IEEE Access*, vol. 7, pp. 60134–60149, 2019, doi: 10.1109/ACCESS.2019.2914999.

[28] B. Wang, C. Li, V. Pavlu, and J. Aslam, "A Pipeline for Optimizing F1-Measure in Multi-label Text Classification," *Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018*, pp. 913–918, 2019, doi: 10.1109/ICMLA.2018.00148.

[29] R. Wang and J. Li, "Bayes test of precision, recall, and F1 measure for comparison of two natural language processing models," *ACL 2019 - 57th Annu. Meet. Assoc. Comput. Linguist. Proc. Conf.*, pp. 4135–4145, 2020, doi: 10.18653/v1/p19-1405.

[30] X. Meng, Q. Zhou, J. Hu, L. Shu, and P. Jiang, "A global support vector regression based on sorted K-fold method," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2017-Decem, pp. 2169–2173, 2018, doi: 10.1109/IEEM.2017.8290276.

[31] Y. Jung, "Multiple predicting K-fold cross-validation for model selection," *J. Nonparametr. Stat.*, vol. 30, no. 1, pp. 197–215, 2018, doi: 10.1080/10485252.2017.1404598.

[32] S. Khodabandehlou and M. Zivari Rahman, *Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior*, vol. 19, no. 1–2. 2017.

[33] A. J. Christy, A. Umamakeswari, L. Priyatharsini, and A. Neyaa, "RFM ranking – An effective approach to customer segmentation," *J. King Saud Univ. - Comput. Inf. Sci.*, 2018, doi: 10.1016/j.jksuci.2018.09.004.

[34] G. Tzanos, C. Kachris, and D. Soudris, "Hardware Acceleration on Gaussian Naive Bayes Machine Learning Algorithm," *2019 8th Int. Conf. Mod. Circuits Syst. Technol. MOCAST 2019*, pp. 1–5, 2019, doi: 10.1109/MOCAST.2019.8741875.

[35] M. Ontivero-Ortega, A. Lage-Castellanos, G. Valente, R. Goebel, and M. Valdes-Sosa, "Fast Gaussian Naïve Bayes for searchlight classification analysis," *Neuroimage*, vol. 163, pp. 471–479, 2017, doi: 10.1016/j.neuroimage.2017.09.001.