Prediksi *Customer Churn* menggunakan Teknik Machine Learning

I. *Introduction*

Banyaknya persaingan antar perusahaan saat ini menjadi tantangan utama yang harus di hadapi oleh sebuah perusahaan. Berbagai inovasi dan strategi khusus perlu dilakukan agar perusahaan memiliki langkah yang lebih maju dibandingkan perusahaan lainnya dan tetap bisa bertahan. Ada banyak faktor yang dapat meningkatkan penjualan produk suatu perusahaan, salah satunya yaitu pelanggan.

Chu (dalam KhakAbi *et al*. 2010) mengatakan untuk memperoleh pelanggan baru memerlukan biaya hingga 10 kali lipat lebih mahal dibandingkan biaya untuk mempertahankan pelanggan yang ada. Berdasarkan fakta tersebut, tentunya perusahaan akan lebih memilih mempertahankan pelanggan lama dan menghindari *churn* pelanggan. Maka dari itu, perusahaan perlu untuk memprediksi pelanggannya agar dapat mengetahui tingkat loyalitas pelanggan dengan melakukan prediksi *customer churn*.

Customer Churn atau yang dikenal juga dengan pindahnya pelanggan adalah pemutusan jasa suatu perusahaan oleh pelanggan karena pelanggan tersebut memilih menggunakan jasa layanan lain (Masarifoglu & Buyuklu, 2019). Dengan menggunakan prediksi *churn*, dapat mengidentifikasi *customer churn* sejak awal sebelum mereka berpindah ke perusahaan lain. Hal ini juga dapat membantu sektor CRM (*Customer Relationship Management*) agar dapat mempertahankan *customer*, sehingga mengurangi potensi kerugian yang dialami perusahaan (Rachmi, 2020).

Pada project ini terdapat beberapa inputan dari permasalahan yang ada yaitu *age*, *days_since_last_login*, *points_in_wallet*, *gender*, *region_category*, *membership_category*, *joined_through_referral*, *preferred_offer_types*, *medium_of_operation*, *internet_option*, *used_special_discount*, *offer_application_preference, past_complaint, complaint_status* dan *feedback*. Selanjutnya untuk melakukan prediksi *customer churn* digunakan beberapa metode klasifikasi seperti metode *Logistic Regression, Naïve Bayes, Decision Tree, SVM* dan *Random Forest*. Sedangkan untuk mengetahui model prediksi tergolong bagus atau tidak dilakukan pengujian ketepatan prediksi menggunakan kurva ROC dan *confusion matrix*.

Tujuan project ini diharapkan dapat membantu perusahaan dalam mengambil keputusan atau tindakan saat adanya pelanggan yang terdeteksi *churn*.

II. *Related Work*

Penelitian tentang prediksi *churn*, sudah dilakukan sebelumnya oleh Olivia *et al.* (2015) dengan melakukan Analisis Prediksi *Churn* Menggunakan Metode *Logistic Regression* dan Algoritma *Decision Tree*. Penelitian tersebut membandingkan antara metode *Decision Tree* dan *Logistic Regression* sehingga didapatkan hasil bahwa *Decision Tree* menghasilkan performansi lebih baik dibandingkan *Logistic Regression* dengan nilai akurasi 94,42% dan waktu 0,064 *second*. Sedangkan performansi yang dihasilkan metode *Logistic Regression* dengan akurasi sebesar 80,73% dan waktu 0,935 *second*. Hal yang sama juga dilakukan pada *project* ini yaitu melakukan prediksi *churn* menggunakan metode *logistic regression* dan *decision tree*. Perbedaannya terletak

pada banyaknya metode yang akan dibandingkan untuk mendapatkan performa model yang terbaik.

Dhea (2022) telah melakukan penelitian tentang prediksi *churn* dengan membuat Implementasi *Churn Prediction* Di Industri Telekomunikasi Dengan Metode *Logistic Regression* dan *Correlation-Based Feature Selection*. Tujuan penelitiannya untuk mempertahankan pelanggan didalam *Customer Relationship Management* (CRM). Hasil penelitian yang didapatkan yaitu tingkat akurasi prediksi sebesar 85,75%. Persamaan penelitian ini dengan *project* yang akan dilakukan yaitu melakukan implementasi *churn prediction* menggunakan *logistic regression* sedangkan perbedaannya yaitu pada project ini akan dilakukan perbandingan beberapa metode klasifikasi seperti metode *Logistic Regression, Naïve Bayes, Decision Tree, SVM* dan *Random Forest*.

Velia (2022) juga sebelumnya telah melakukan penelitian dengan membuat Klasifikasi *Customer Churn* Pada Perusahaan Telekomunikasi Menggunakan *Support Vector Machine* dimana dalam pengembangannya untuk mendapatkan performa model yang lebih baik dilakukan optimalisasi terhadap parameter SVM menggunakan *Manual Search* dengan *Trial-and-Error*, dengan matriks evaluasi yang telah disesuaikan dan dapat diadaptasi sesuai dengan strategi yang akan digunakan. Penelitian ini memfokuskan pengimplementasian SVM terhadap klasifikasi *customer churn* dengan melakukan optimalisasi parameter SVM. Sedangkan *project* yang akan dilakukan berfokus pada performa model dengan membandingkan beberapa metode.

III. *Dataset & Features*

Dataset yang digunakan pada project ini berasal dari Predict the churn risk rate | Practice Problems (hackerearth.com). Dataset tersebut memiliki 36.992 *record* dengan 23 variabel. Berikut informasi dari variable data yang digunakan :

- *age* (umur pelanggan)
- *gender* (jenis kelamin pelanggan) (Object: "F", "M")
- *security_no* (nomor keamanan yang unik untuk mengidentifikasi seseorang)
- *region_category* (wilayah tempat pelanggan) (Object: "Village", "City", "Town")
- *membership_category* (kategori keanggotaan yang digunakan pelanggan) (Object: "Basic Membership", "Silver Membership", "Platinum Membership", "Gold Membership", "No Membership", "Premium Membership")
- *joining_date* (tanggal ketika pelanggan menjadi anggota)
- *joined_through_referral* (apakah pelanggan saat bergabung menggunakan kode atau ID rujukan) (Object: "Yes", "No")
- *referral_id* : (id referral)
- *preferred_offer_types* (jenis penawaran yang disukai pelanggan) (Object : "Gift Vouchers/Coupons", "Credit/Debit Card Offers", "Without Offers")
- *medium_of_operation* (media yang digunakan pelanggan untuk bertransaksi) (Object : "Desktop", "Smartphone", "Both")
- *internet_option* (layanan internet yang digunakan pelanggan) (Object: "Wi-Fi", "Mobile_Data", "Fiber_Optic")
- *last_visit_time* (terakhir kali pelanggan mengunjungi website)
- *days_since_last_login* (merepresentasikan hari sejak pelanggan terakhir mengunjungi situs web)
- *avg_time_spent* (rata-rata waktu yang dihabiskan oleh pelanggan di situs web)
- *avg_transaction_value* (rata-rata transaksi yang dilakukan oleh pelanggan)
- *avg_frequency_login_days* (rata-rata pelanggan *login* di situs web)

- *points_in_wallet* (poin yang diberikan pelanggan pada setiap transaksi)
- *used_special_discount* (apakah pelanggan menggunakan diskon yang ditawarkan) (Object: "Yes ", "No")
- *offer_application_preference* (apakah pelanggan suka penawaran yang direkomendasikan) (Object: "Yes ", "No")
- *past_complaint* (apakah pelanggan mengajukan keluhan) (Object: "Yes ", "No")
- *complaint_status* (keluhan yang diajukan oleh pelanggan diselesaikan) (Object: "Not Applicable", "Unsloved", "Solved", "Solved in Follow-up", "No Information Available")
- *feedback* (umpan balik yang diberikan pelanggan) (Object : "Poor Product Quality", "No reason specified", "Too many ads", "Poor Website", "Poor Customer Service", "Reasonable Price", "User Friendly Website", "Products always in Stock", "Quality Customer Care")
- *churn_risk_score* : (merepresentasikan churn/no churn) (Integer : "1", "0")

Untuk lebih jelasnya, dataset dapat dilihat pada gambar berikut :

	age	gender	security_no	region_category	membership_category	joining_date	joined_through_referral	referral_id	preferred_offer_types	medium_of_operatio
0	18	F	XW0DQ7H	Village	Platinum Membership	2017-08-17	No	xxxxxxxx	Gift Vouchers/Coupons	
1	32	F	5K0N3X1	City	Premium Membership	2017-08-28	?	CID21329	Gift Vouchers/Coupons	Deskto
2	44	F	1F2TCL3	Town	No Membership	2016-11-11	Yes	CID12313	Gift Vouchers/Coupons	Deskto
3	37	M	VJGJ33N	City	No Membership	2016-10-29	Yes	CID3793	Gift Vouchers/Coupons	Deskto
4	31	F	SVZXCWB	City	No Membership	2017-09-12	No	xxxxxxxx	Credit/Debit Card Offers	Smartphon

5 rows × 23 columns

Gambar 1. Dataset Dalam Bentuk Dataframe

Sedangkan untuk melihat tipe data yang ada pada dataset, dapat dilihat pada gambar berikut:

```
Data columns (total 23 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   age                          36992 non-null  int64
 1   gender                       36992 non-null  object
 2   security_no                  36992 non-null  object
 3   region_category              31564 non-null  object
 4   membership_category          36992 non-null  object
 5   joining_date                 36992 non-null  object
 6   joined_through_referral      36992 non-null  object
 7   referral_id                  36992 non-null  object
 8   preferred_offer_types        36704 non-null  object
 9   medium_of_operation          36992 non-null  object
 10  internet_option              36992 non-null  object
 11  last_visit_time              36992 non-null  object
 12  days_since_last_login        36992 non-null  int64
 13  avg_time_spent               36992 non-null  float64
 14  avg_transaction_value        36992 non-null  float64
 15  avg_frequency_login_days     36992 non-null  object
 16  points_in_wallet             33549 non-null  float64
 17  used_special_discount        36992 non-null  object
 18  offer_application_preference 36992 non-null  object
 19  past_complaint               36992 non-null  object
 20  complaint_status             36992 non-null  object
 21  feedback                     36992 non-null  object
 22  churn_risk_score             36992 non-null  int64
```

Gambar 2. Tipe Data Pada Dataset

Setelah pendefinisian data telah berhasil dilakukan, kolom dataset dipisah menjadi dua bagian yaitu *input data* dan *output data*. Kemudian bagi data tersebut menjadi *data train, data valid* dan *data test*. Banyaknya baris dan kolom pada *data training, data validation* dan *data test* dapat dilihat pada gambar di bawah ini:

```
X_train shape: (25894, 22)
X_test shape: (5549, 22)
X_valid shape:(5549, 22)
y_train shape: (25894,)
y_test shape: (5549,)
y_valid shape: (5549,)
```

Gambar 3. *Shape* Pada Data Training/Validation/Test

Pada tahap *pre-processing data*, deklarasikan variabel untuk memisahkan kolom berupa *categorical*, *numeric* dan juga *drop* kolom seperti gamber berikut :

```
categorical_column = ["gender", "region_category", "membership_category",
                      "joined_through_referral", "preferred_offer_types",
                      "medium_of_operation", "internet_option",
                      "used_special_discount", "offer_application_preference",
                      "past_complaint","complaint_status","feedback"]

numerical_column = ["age", "days_since_last_login", "points_in_wallet"]

drop_column = ["security_no", "joining_date", "referral_id", "last_visit_time", "avg_time_spent",
                      "avg_transaction_value", "avg_frequency_login_days",]
```

Gambar 4. Pendeklarasian Kolom *Categorical/Numeric/Drop*

Dikarenakan pada *project* ini tidak berfokus pada *timeseries* maka kolom yang berkaitan dengan waktu/tanggal akan di masukkan ke dalam variabel *drop* kolom dan akan dilakukan penghapusan terhadap kolom-kolom tersebut.

Selanjutnya lakukan pengecekan nilai yang terdapat pada *numerical* kolom. Kolom "*days_since_last_login*" dan "*points_in_wallet*" memiliki nilai negatif. Ubah nilai negatif tersebut menjadi "nan" agar nantinya dapat dilakukan *imputation numerical* menggunakan nilai median. Pengubahan nilai tersebut juga dilakukan pada *data valid* dan juga *data test*. Setelah *numerical column* selesai dilakukan pengecekan, maka lakukan hal yang sama terhadap *categorical column*. Ubah nilai "unknown" dan "?" yang ada pada kolom "*gender*", "*joined_through_referral*", "*medium_of_operation*" menjadi "nan" agar dapat dilakukan *imputation categorical* menggunakan nilai "KOSONG". Pengubahan nilai tersebut juga dilakukan pada *data valid* dan juga *data test*.

Agar dapat dilakukan proses modeling, ubah *categorical column* menjadi nilai numerik dengan menggunakan *one hot encoder* kemudian gabungkan antara *numerical column* dan *categorical column*. Lakukan hal yang sama pada *data valid* dan juga *data test*. Hasil dari pengubahan dan penggabungan tersebut dapat dilihat pada gambar berikut :

	age	days_since_last_login	points_in_wallet	gender_F	gender_M	region_category_City	region_category_KOSONG	region_category_Town	region_category_V
0	40.0	13.0	548.870000	0.0	1.0	0.0	1.0	0.0	
1	49.0	17.0	773.760000	0.0	1.0	0.0	1.0	0.0	
2	46.0	12.0	353.290489	1.0	0.0	0.0	1.0	0.0	
3	40.0	13.0	797.180000	0.0	1.0	1.0	0.0	0.0	
4	48.0	14.0	758.740000	0.0	1.0	0.0	1.0	0.0	

Gambar 5. *Categorical OHE* dan Penggabungan Kolom

Untuk meningkatkan akurasi model dan membuat data tahan terhadap outlier, tahap normalisasi data sangat penting. Pada *project* ini dilakukan *StandardScaler* yang bertujuan untuk membuat rata-rata 0 dan variansi 1. Hasil dari normalisasi data dapat dilihat pada gambar berikut:

	age	days_since_last_login	points_in_wallet	gender_F	gender_M	region_category_City	region_category_KOSONG	region_category_Town
0	-1.206558	-0.512906	0.268487	-0.997489	0.997489	-0.721973	-0.414310	1.270711
1	0.116142	1.698067	-2.474832	-0.997489	0.997489	-0.721973	2.413653	-0.786961
2	-0.198787	-0.144410	-0.686999	1.002518	-1.002518	1.385093	-0.414310	-0.786961
3	1.186899	0.961076	0.144788	-0.997489	0.997489	-0.721973	-0.414310	-0.786961
4	-0.891630	-0.512906	-0.108614	1.002518	-1.002518	-0.721973	-0.414310	-0.786961
...	...	...	...	...	...	...	...	...
25889	1.312870	-1.249897	-1.589980	1.002518	-1.002518	-0.721973	-0.414310	1.270711
25890	1.186899	-1.802640	0.481441	1.002518	-1.002518	-0.721973	-0.414310	1.270711
25891	-1.080587	-0.512906	0.043023	1.002518	-1.002518	-0.721973	2.413653	-0.786961
25892	1.249885	0.592581	0.018620	-0.997489	0.997489	-0.721973	-0.414310	1.270711
25893	1.501828	-1.065649	0.506910	-0.997489	0.997489	-0.721973	-0.414310	1.270711

Gambar 6. Normalisasi Data

Selanjutnya lakukan pengecekan terhadap target data, untuk melihat apakah data *balance* atau *imbalance*.

```
1    0.541027
0    0.458973
```

Gambar 7. Jumlah Target Data

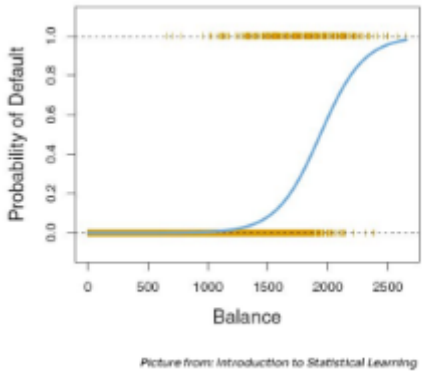
Dari gambar tersebut, dapat diketahui bahwa data yang dimiliki adalah *balance*. Sehingga saat melakukan pemodelan, acuan dasar untuk memastikan performa model sudah berjalan dengan baik yaitu dengan melihat nilai dari *baseline model*. Performa model nantinya harus melebihi dari nilai *baseline model*. *Baseline model* klasifikasi berasal dari proporsi kelas terbesar. Maka dari itu, *baseline* pada *project* ini adalah 54%.

IV. *Metode*

Pada *project* ini dilakukan *modeling* pada beberapa metode yaitu *Logistic Regression, Naïve Bayes, Decision Tree, SVM* dan *Random Forest*.

1. *Logistic Regression*

Logistic regression melakukan prediksi pada nilai output yang merentang dari 0-1 sehingga dapat menekan nilai-nilai yang negatif atau positif. Konsep *probability* tidak digunakan saat melakukan klasifikasi dengan *logistic regression*, hanya melihat dimana nilai titik saat dimasukkan ke dalam fungsi persamaan garis pembagi.



Gambar 8. Ilustrasi *Logistic Regression*

2. *Naïve Bayes*

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan.

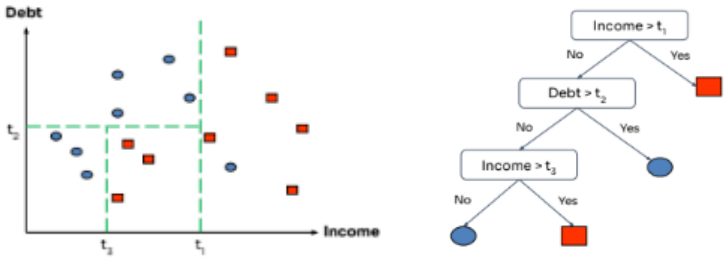
Persamaan dari *naïve bayes* adalah :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \qquad (1)$$

Di mana :
X :Data dengan *class* yang belum diketahui
H : Hipotesis data merupakan suatu *class* spesifik
$P(H|X)$:Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)
$P(H)$: Probabilitas hipotesis H (prior probabilitas)
$P(X|H)$:Probabilitas X berdasarkan kondisi pada hipotesis H
$P(X)$: Probabilitas X

3. *Decision Tree*

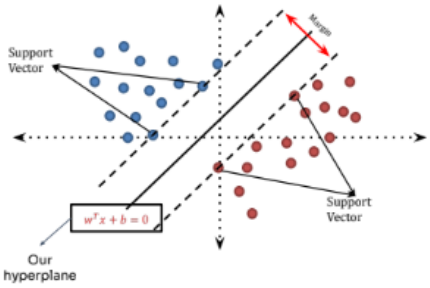
Decision Tree merupakan salah satu model yang cukup mudah di interpretasikan karena dikelompokkan per kategori kolom/*unlimited if-else*. Secara umum, *decision tree* membagi dataset menjadi beberapa *region* dan pencarian hasil prediksinya berdasarkan data yang ada di *region* tersebut. *Decision tree* bisa membuat *logic tree*. Jika logic tree berhasil dibuat maka akan lebih mudah melakukan interpretasi dari model yang ada.



Gambar 9. Ilustrasi *Decision Tree*

4. *Support Vector Machine (SVM)*

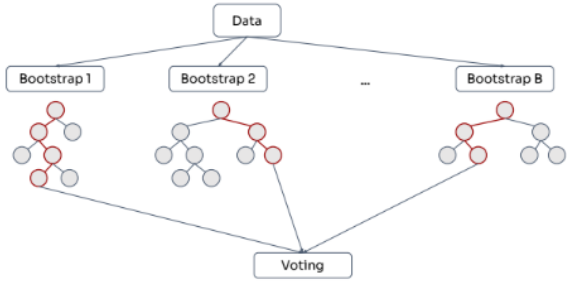
SVM dapat membuat garis *boundary* pada data. SVM membuat garis dari vektor terluar kelasnya yang bisa memaksimalkan *margin* antar kelas. Makin besar *margin* maka akan mengurangi potensi kesalahan dalam melakukan prediksi. Akurasi dari model SVM bergantung pada pemilihan parameter *kernel*, karena parameter ini memiliki dampak yang signifikan terhadap performa dari metode *kernel*. Banyaknya parameter bergantung pada *margin* yang memisahkan beberapa kumpulan data.



Gambar 10. Ilustrasi SVM

5. *Random Forest*

Konsep dari *random forest* mirip seperti bagging, perbedaannya terletak pada pembuatan *decision tree*. Jika *decision tree* dengan konsep *bagging* menggunakan keseluruhan fitur yang ada, *random forest* hanya menggunakan beberapa fitur agar model *decision tree* saling tidak berkorelasi antara satu dan lainnya. Setelah proses tersebut telah dilakukan, akhir dari prediksinya adalah hasil agregasi dari setiap *prediksi decision* tree yang sudah dibuat.



Gambar 11. Ilustrasi *Random Forest*

V. *Experiments/Results/Discussion*

Data *validation* yang telah dibangun akan dilakukan tahap pengujian yang meliputi nilai akurasi, *precision*, *recall* dan *f1-score*. Nilai akurasi didapatkan dari prediksi benar untuk data positif dan negatif dari keseluruhan data. Akurasi dapat menggambarkan keakuratan model klasifikasi yang digunakan. Nilai akurasi dapat diperoleh menggunakan persamaan berikut ini.

$$accuracy\ (\%) = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

Precision adalah rasio yang memiliki prediksi nilai benar positif jika dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Precision* dapat menggambarkan keakuratan data yang diinginkan dengan hasil prediksi yang diperoleh model klasifikasi. Nilai *precision* dapat diperoleh menggunakan persamaan berikut ini.

$$precision = \frac{TP}{TP + FP}$$

Recall adalah efektivitas dari pengklasifikasi dalam mengidentifikasi label positif. Nilai *Recall* dapat diperoleh menggunakan persamaan berikut ini.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score adalah nilai yang menandakan jika model yang dibangun memiliki nilai *precision* dan *recall* yang baik. Nilai *F1 score* dapat diperoleh menggunakan persamaan berikut ini.

$$\frac{1}{F1} = \frac{1}{2}\left(\frac{1}{precision} + \frac{1}{recall}\right)$$

Nilai akurasi, *precision*, *recall* dan *f1-score* dapat diperoleh menggunakan *confusion matrix*. Berikut ini hasil *confusion matrix* yang didapatkan pada setiap metode.

1. *Logistic Regression*

Pada metode *logistic regression, hyperparameter* terbaik terjadi pada saat 'C': 0.02636 dan 'penalty': 'l1'. Hasil evaluasi performa dari model *logistic regression* dapat dilihat pada *confusion matrix* dan *classification report* berikut.

```
[2362,  181]
[ 300, 2698]
```

	precision	recall	f1-score	support
0	0.89	0.93	0.91	2543
1	0.94	0.90	0.92	2998
accuracy			0.91	5541
macro avg	0.91	0.91	0.91	5541
weighted avg	0.91	0.91	0.91	5541

Gambar 12. *Confusion Matrix* dan *Classification Report Logistic Regression*

2. *Naïve Bayes*

Pada metode *naïve bayes, hyperparameter* terbaik terjadi pada saat 'var_smoothing': 0.03511. Hasil evaluasi performa dari model *naïve bayes* dapat dilihat pada *confusion matrix* dan *classification report* berikut.

```
[1644,  899]
[   0, 2998]
```

	precision	recall	f1-score	support
0	1.00	0.65	0.79	2543
1	0.77	1.00	0.87	2998
accuracy			0.84	5541
macro avg	0.88	0.82	0.83	5541
weighted avg	0.88	0.84	0.83	5541

Gambar 13. *Confusion Matrix* dan *Classification Report Naïve Bayes*

3. *Decision Tree*

Pada metode *decision tree, hyperparameter* terbaik terjadi pada saat '*max_depth*': 15, '*max_features*': 'log2', dan '*min_samples_split*': 2. Hasil evaluasi performa dari model *naïve bayes* dapat dilihat pada *confusion matrix* dan *classification report* berikut.

```
[2158,  385]
[ 311, 2687]
              precision    recall  f1-score   support

           0       0.87      0.85      0.86      2543
           1       0.87      0.90      0.89      2998

    accuracy                           0.87      5541
   macro avg       0.87      0.87      0.87      5541
weighted avg       0.87      0.87      0.87      5541
```

Gambar 14. *Confusion Matrix* dan *Classification Report Decision Tree*

4. *Support Vector Machine (SVM)*

Pada metode *SVM*, *hyperparameter* terbaik terjadi pada saat 'C': 1, dan '*kernel*': '*rbf*'. Hasil evaluasi performa dari model SVM dapat dilihat pada *confusion matrix* dan *classification report* berikut.

```
[2228,  315]
[ 384, 2614]
              precision    recall  f1-score   support

           0       0.85      0.88      0.86      2543
           1       0.89      0.87      0.88      2998

    accuracy                           0.87      5541
   macro avg       0.87      0.87      0.87      5541
weighted avg       0.87      0.87      0.87      5541
```

Gambar 15. *Confusion Matrix* dan *Classification Report SVM*

5. *Random Forest*

Pada metode *Random Forest*, *hyperparameter* terbaik terjadi pada saat '*criterion*': '*gini*', '*max_depth*': 5, '*max_features*': '*log2*', dan '*n_estimators*': 300. Hasil evaluasi performa dari model r andom forest dapat dilihat pada confusion matrix dan classification report berikut.

```
[2311,  232]
[ 163, 2835]
              precision    recall  f1-score   support

           0       0.93      0.91      0.92      2543
           1       0.92      0.95      0.93      2998

    accuracy                           0.93      5541
   macro avg       0.93      0.93      0.93      5541
weighted avg       0.93      0.93      0.93      5541
```

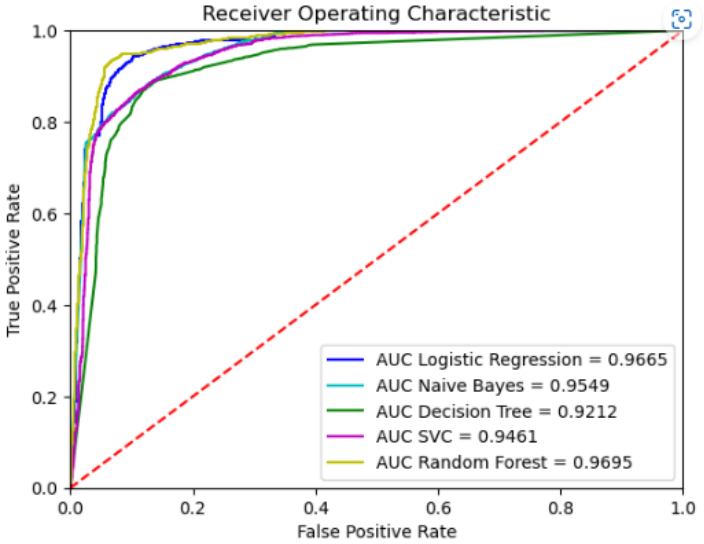
Gambar 16. Confusion Matrix dan Classification Report Random Forest

Perbandingan hasil dari *confusion matrix* dan *classification report* pada *data validation* dan juga *data testing* dapat dilihat tabel berikut.

Metode	Data Validation	Data Testing
Logistic Regression	precision recall f1-score support 0 0.89 0.93 0.91 2543 1 0.94 0.90 0.92 2998 accuracy 0.91 5541 macro avg 0.91 0.91 0.91 5541 weighted avg 0.91 0.91 0.91 5541	precision recall f1-score support 0 0.88 0.93 0.91 2543 1 0.94 0.89 0.92 3000 accuracy 0.91 5543 macro avg 0.91 0.91 0.91 5543 weighted avg 0.91 0.91 0.91 5543

Metode	Data Validation	Data Testing
Naïve Bayes	precision recall f1-score support 0 1.00 0.65 0.79 2543 1 0.77 1.00 0.87 2998 accuracy 0.84 5541 macro avg 0.88 0.82 0.83 5541 weighted avg 0.88 0.84 0.83 5541	precision recall f1-score support 0 1.00 0.65 0.78 2543 1 0.77 1.00 0.87 3000 accuracy 0.84 5543 macro avg 0.88 0.82 0.83 5543 weighted avg 0.87 0.84 0.83 5543
Decision Tree	precision recall f1-score support 0 0.87 0.85 0.86 2543 1 0.87 0.90 0.89 2998 accuracy 0.87 5541 macro avg 0.87 0.87 0.87 5541 weighted avg 0.87 0.87 0.87 5541	precision recall f1-score support 0 0.86 0.87 0.86 2543 1 0.89 0.88 0.88 3000 accuracy 0.87 5543 macro avg 0.87 0.87 0.87 5543 weighted avg 0.87 0.87 0.87 5543
SVM	precision recall f1-score support 0 0.85 0.88 0.86 2543 1 0.89 0.87 0.88 2998 accuracy 0.87 5541 macro avg 0.87 0.87 0.87 5541 weighted avg 0.87 0.87 0.87 5541	precision recall f1-score support 0 0.85 0.89 0.87 2543 1 0.90 0.87 0.88 3000 accuracy 0.88 5543 macro avg 0.88 0.88 0.88 5543 weighted avg 0.88 0.88 0.88 5543
Random Forest	precision recall f1-score support 0 0.93 0.91 0.92 2543 1 0.92 0.95 0.93 2998 accuracy 0.93 5541 macro avg 0.93 0.93 0.93 5541 weighted avg 0.93 0.93 0.93 5541	precision recall f1-score support 0 0.94 0.91 0.93 2543 1 0.93 0.95 0.94 3000 accuracy 0.93 5543 macro avg 0.93 0.93 0.93 5543 weighted avg 0.93 0.93 0.93 5543

Selain menggunakan *confusion matrix*, dilakukan pengujian pada ROC *Curve* pada masing-masing metode. Pengujian ROC *Curve* berisikan *false positif rate* dan *true positif rate*. Semakin kurva mendekati *true positive rate* maka prediksi dapat dikatakan sangat bagus hal ini dikarenakan *rate* untuk salah dalam memprediksi semakin kecil. Hasil ROC *Curve* pada setiap metode dapat dilihat pada gambar berikut.



Gambar 17. ROC *Curve* Pada Setiap Metode

VI. *Conclusion/Future Work*

Pada kasus *churn* lebih besar resiko yang akan didapatkan, jika prediksi menghasilkan *tidak churn* padahal kenyataannya *churn*. Dengan situasi tersebut maka fokus *project* ini yaitu memperkecil nilai *false negative* yang berarti memperbesar nilai *recall*. Dari pengujian *confusion matrix* dan pengujian ROC *curve* yang telah dilakukan antara metode *logistic regression, naïve bayes, decision tree, SVM* dan *random forest* didapatkan hasil bahwa metode terbaik terdapat pada metode *random forest*. Dengan nilai *recall* sebesar 93% dan nilai AUC sebesar 0.9695. Untuk meningkatkan performa model yang lebih baik, teknik *boosting* atau *stacking* dapat digunakan pada eksplorasi selanjutnya.

Link Github :

https://github.com/uyunmubarak/churn-prediction

Reference

S. KhakAbi, M. R. Gholamian, and M. Namvar, "Data Mining Applications in Customer Churn Management," 2010 International Conference on Intelligent Systems, Modelling and Simulation, pp. 220–225, Jan. 2010

Masarifoglu, M., & Buyuklu, A. H. (2019). Applying Survival Analysis to 71 Telecom Churn Data. American Journal of Theoretical and Applied Statistics, 8(6), 261–275. https://doi.org/10.11648/j.ajtas.20190806.18

Cici Olivia, Indwiarti, Sibaroni Yulian (2015). Analisis Prediksi Churn Menggunakan Metode Logistic Regression dan Algoritma Decision Tree.Jurnal e-Proceeding of Engineering, Volume 2, Nomor 2.

Adhelia Nurfira Rachmi, (2020). Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn.

Dhea Laksnu Prianto, IIn Ernawati, Nurul Chamidah, (2022). Implementasi Churn Prediction Di Industri Telekomunikasi Dengan Metode Logistic Regression Dan Correlation-Based Feature Selection. Seminar Nasional Informatika, Sistem Informasi dan Keamanan Siber (SEINASI-KESI). Jakarta-Indonesia, Januari 2022.

Amanda Velia, R. S. Theodorus, N. S. Sandi, A. P. Anindya, Fajar Indrayatna. (2022). Klasifikasi Customer Churn Pada Perusahaan Telekomunikasi Menggunakan Support Vector Machine. Seminar Nasional Statistika Aktuaria I (2022).