MODUL 3
KLASIFIKASI PADA DATASET TAK SEIMBANG
(Minggu ke 4 – 5)

1. TUJUAN

Tujuan praktikum ini adalah mahasiswa mampu:

1) mampu memahami konsep beberapa metode klasifikasi, utamanya klasifikasi biner;
2) mampu memahami dan menerapkan under-sampling, over-sampling, kombinasi over dan under-sampling, serta SMOTE over-sampling;
3) mampu melakukan perbandingan dan evaluasi kinerja klasifikasi dengan menggunakan beberapa metode resampling; dan
4) membangun sistem/aplikasi klasifikasi dengan Jupyter Notebook.

2. DASAR TEORI

Regresi Logistik Biner

Menurut Harlan dalam (Brahmantyo et al., 2021), Regresi Logistik Biner digunakan untuk mengukur pengaruh variabel independen kontinu atau kategoris (X) terhadap dependen variabel (Y) yang memiliki nilai dikotomis atau biner. Bentuk umum regresi logistik:

$$logit(Y) = lnO(Y) = ln\frac{P(Y)}{1-P(Y)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p \quad (1)$$

yang mana dapat ditulis dalam bentuk Odds dari Y dengan:

$$O(Y) = \frac{P(Y)}{1-P(Y)} = e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p} \quad (2)$$

maka dari persamaan (2) dapat dibentuk menjadi persamaan probabilistik dalam regresi logistik:

$$P(Y) = \frac{e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}} \quad (3)$$

di mana persamaan (3) dapat digunakan untuk menghitung nilai probabilitas dalam model regresi logistik. Estimasi parameter regresi logistik untuk variabel tak bebas yang dikotomis atau biner, menggunakan Distribusi Bernoulli sebagai distribusi probabilitas. Ini akan menghasilkan fungsi likelihood yang merupakan kombinasi dari semua parameter fungsi probabilitas dalam model. Oleh karena itu, dapat dikatakan bahwa fungsi likelihood untuk regresi logistik adalah:

$$L(\beta) = \prod_{i=1}^{n} [P(Y)]^{Y_i} . [1 - P(Y)]^{1 - Y_i}, Y_i = \{0, 1\} \quad (4)$$

maka ketika persamaan (3) disubstitusikan ke persamaan (4), maka akan menghasilkan:

$$L(\beta) = \prod_{i=1}^{n} \left[e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}\right]^{Y_i} . \left[1 + e^{\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p}\right] \quad (5)$$

dan jika persamaan (5) ditransformasikan dengan logaritma natural, menjadi:

$$\ln L(\beta) = \sum_{i=1}^{n} \left[Y_i \left\{ \sum_{p=0}^{n} \beta_p X_{ip} - \ln \left(1 + e^{\sum_{p=0}^{p} \beta_p X_{ip}}\right) \right\} \right] \quad (6)$$

Adapun pada persamaan (6) akan diestimasi dengan pendekatan Newton Raphson, karena fungsi tidak dalam bentuk linier dan perlu pendekatan untuk memperkirakan nilai parameter.

Menurut Hosmer dan Lemeshow dalam (Chairunnisa et al., 2017), untuk menguji signifikansi dari parameter dalam model digunakan uji rasio likelihood dan uji Wald. Uji rasio likelihood digunakan untuk mengetahui apakah variabel prediktor secara bersama-sama mempengaruhi respon. Hipotesis dalam uji rasio likelihood yaitu

$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0$

H_1 : paling sedikit ada satu $\beta_j \neq 0$ dengan j = 1,2,…,p

Statistik uji rasio likelihood adalah

$$\boldsymbol{G = -2\, ln \left(\frac{likelihood\ tanpa\ variabel\ bebas}{likelihood\ dengan\ variabel\ bebas}\right)} \quad (7)$$

Kriteria uji yaitu H0 ditolak jika $G > \chi^2(\alpha, p)$.

Sedangkan uji Wald dilakukan untuk mengetahui signifikansi parameter terhadap variabel respon. Hipotesis uji Wald yaitu:

$H_0 : \beta_j = 0$ dengan j = 1,2, …, p

$H_1 : \beta_j \neq 0$ dengan j = 1,2,…,p

Statistik uji Wald adalah

$$W_j = \left\{\frac{\hat{\beta}_j}{se\,(\hat{\beta}_j)}\right\}^2 \quad (8)$$

Kriteria uji yaitu H_0 ditolak jika $W_j > \chi^2(\alpha, 1)$.
Model yang diperoleh diuji kesesuaiannya, uji kesesuaian model digunakan untuk mengetahui apakah model efektif dalam menjelaskan variabel hasil. Hipotesis yang digunakan dalam uji kesesuaian model yaitu:
H_0 : Model sesuai (Nilai observasi sama dengan nilai prediksi)
H_1 : Model tidak sesuai (Nilai observasi tidak sama dengan nilai prediksi)
Statistik uji yang digunakan adalah:

$$\hat{C} = \sum_{k=1}^{g} \frac{(O_k - n\bar{\pi}_k)^2}{n_k \pi_k (1 - \pi_k)} \quad (9)$$

Kriteria uji yang digunakan yaitu tolak H_0 jika $\hat{C} > \chi^2(\alpha, g - 2)$.

3. STUDI KASUS

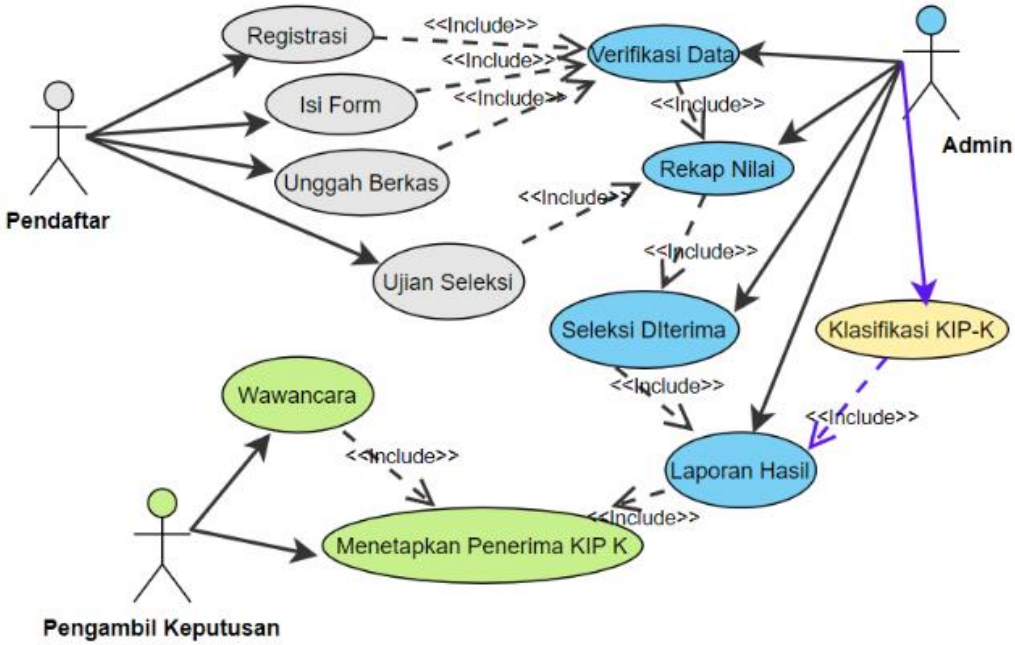
Dalam praktikum ini, focus untuk meyelesaikan satu studi kasus, yaitu klasifikasi mahasiswa penerima Kartu Indonesia Pintar Kuliah (KIP Kuliah).

3.1 Latar Belakang

Pada tahun 2020 melalui Program Indonesia Pintar atau PIP, pemerintah telah memberikan bantuan pendidikan bagi 200 ribu mahasiswa yang diterima di perguruan tinggi termasuk penyandang disabilitas dalam bentuk Kartu Indonesia Pintar Kuliah (KIP Kuliah). Pada tahun 2021, pemerintah melalui Pusat Layanan Pembiayaan Pendidikan (Puslapdik) Kemendikbud kembali menyalurkan bantuan untuk melanjutkan pendidikan tinggi kepada 200 ribu mahasiswa penerima KIP Kuliah baru, selain terus menjamin penyaluran KIP Kuliah *on going* dan Bidikmisi *on going* sampai masa studi selesai. Penerima KIP Kuliah adalah Siswa Sekolah Menengah Atas (SMA), Sekolah Menengah Kejuruan (SMK), atau bentuk lain yang sederajat yang akan lulus pada tahun berjalan atau lulus 2 (dua) tahun sebelumnya, memiliki potensi akademik baik tetapi memiliki keterbatasan ekonomi (Kemendikbud, 2021).

Seleksi Bersama Masuk Politeknik Negeri atau disingkat SBMPN merupakan jalur masuk Politeknik Negeri melalui tes tulis yang bertujuan untuk memilih calon mahasiswa yang mempunyai kemampuan akademik untuk mengikuti dan menyelesaikan pendidikan di politeknik sesuai dengan batas waktu yang telah ditetapkan. Bagi pendaftar yang kurang mampu dari segi ekonomi

dapat memilih skema KIP Kuliah ini. Gambar 1 merupakan sistem seleksi mahasiswa baru penerima KIP Kuliah.



Gambar 2. Sistem Seleksi KIP Kuliah

Seleksi KIP Kuliah selama ini berdasarkan wawancara. Kasus yang sering terjadi di lapangan adalah beberapa mahasiswa yang mendapatkan KIP Kuliah ternyata mampu secara ekonomi. Sebaliknya, pada semester berjalan, banyak mahasiswa yang tidak menerima KIP Kuliah melakukan permohonan keringanan UKT karena masalah ekonomi. Selain alas an tersebut, jumlah pendaftar KIP Kuliah cenderung lebih banyak dibandingkan dengan jumlah kuota yang diterima. Oleh sebab itu pihak institusi perlu membuat rumusan klasifikasi sebagai acuan dalam menetapkan mahasiswa baru penerima KIP Kuliah.

3.2 Tujuan

Tujuan dari penelitian ini adalah membuat model klasifikasi penerima KIP Kuliah menggunakan metode Regresi Logistik. Beberapa alternatif model klasifikasi akan dievaluasi kinerjanya untuk dipilih sebagai model terbaik yang dapat dipertimbangkan sebagai rumusan sistem seleksi mahasiswa baru penerima KIP Kuliah.

3.3 Penelitian Terdahulu

Analisis Regresi Logistik Biner digunakan untuk mengklasifikasi penderita hipertensi berdasarkan kebiasaan merokok di RSU Mokopindo Toli-Toli. Ada enam faktor yang diduga berpengaruh pada

penderita hipertensi. Dari hasil penelitian menunjukkan bahwa faktor-faktor yang signifikan mempengaruhi penderita hipertensi adalah lama merokok, jenis rokok yang dihisap, dan cara menghisap rokok. Nilai akurasi klasifikasi sebesar 77,6% (Misna et al., 2018).

Regresi logistik digunakan sebagai model prediktif untuk mengevaluasi probabilitas apakah seorang siswa yang diterima di program sarjana pada Philippine University akan melakukan registrasi atau tidak. Penerapan *machine learning* ini untuk melengkapi keputusan manajemen dan mengestimasi jumlah kelas. Sehingga institusi dapat mengoptimalkan alokasi sumber daya dan memiliki kontrol yang lebih baik atas pendapatan bersih dari uang kuliah (Esquivel & Esquivel, 2020).

Metode klasifikasi Regresi Logistik Biner dan *Regression Tree* (CART) digunakan untuk menentukan karakteristik mahasiswa yang diklasifikasikan menurut dua kategori yaitu: masa studi kurang dari atau sama dengan 5 tahun dan masa studi lebih dari 5 tahun. Faktor-faktor yang mempengaruhi lama belajar siswa berdasarkan metode Regresi Logistik Biner adalah IPK, jenis kelamin, jenis sekolah menengah dan jurusan. Metode Regresi Logistik Biner mampu memprediksi pengamatan secara akurat sebagai sebanyak 75,0%, sedangkan metode CART mampu memprediksi pengamatan secara akurat sebanyak 77,27% (Chairunnisa et al., 2017).

Analisis regresi logistik juga digunakan untuk menentukan faktor-faktor yang mempengaruhi indeks prestasi kumulatif (IPK) mahasiswa FMIPA Universitas Sam Ratulangi Manado. Hasil penelitian ini, program studi dan tempat tinggal berpeluang memiliki pengaruh terhadap IPK (Tampil et al., 2017).

Bayesian Binary Logistic Regression digunakan untuk klasifikasi risiko kematian pasien Covid-19. Hasil penelitian ini menunjukkan bahwa jumlah komorbid berpengaruh signifikan terhadap risiko kematian pasien Covid-19. Semakin banyak jumlah komorbid yang diderita oleh pasien maka semakin tinggi pula risiko kematian pasien tersebut. Adapun ketepatan metode ini dalam mengklasifikasi data sebesar 84,68 % (Shobri et al., 2021). Penerapan model Regresi Logistik Biner juga digunakan untuk mengklasifikasikan saham S&P BSE 30 menjadi dua kategori saham, yaitu kinerja baik atau buruk (Smita, 2021), untuk membuat profil kesehatan (Ambika et al., 2019), dan juga digunakan untuk mengklasifikasikan laporan dari LAPOR! berdasarkan keluhan dan non-keluhan. (Salamah & Ramayanti, 2018).

Dalam penelitian (Khoirunissa et al., 2021) dilakukan perbandingan Regresi Logistik dengan metode *Random Forest,* dan metode *Multilayer Perceptron* untuk klasifikasi penutupan akun pelanggan bank. Dalam penelitian tersebut menggunakan 10.000 responden yang berasal dari Perancis, Spanyol, dan Jerman. Hasil penelitian tersebut menunjukkan bahwa metode *Multilayer Perceptron*

dengan 10 *Fold Cross Validation* adalah model terbaik dengan akurasi 85,5373%.

Metode *Bagging* Regresi Logistik digunakan untuk meningkatkan ketepatan klasifikasi waktu kelulusan mahasiswa STIKOM Bali. *Bagging* Regresi Logistik digunakan untuk melakukan replikasi bootstrap dalam menangani kestabilan pendugaan parameter dan meningkatkan klasifikasi regresi logistik. Metode ini mampu menaikkan ketepatan klasifikasi sebesar 1,01% dari data set tunggal pada replikasi bootstrap 70 kali dengan nilai ketepatan klasifikasi 86,40% (Suniantara et al., 2018).

Kinerja metode Regresi Logistik dan *Support Vector Machine* (SVM) juga dibandingkan untuk tiga data set yang berbeda. Untuk mengevaluasi kinerja dua metode tersebut digunakan *Apparent Error Rate* (Aper) dan statistik Press'Q. Dalam penelitian ini disimpulkan bahwa SVM memiliki kinerja klasifikasi lebih baik daripada Regresi Logistik (Widodo & Handoyo, 2017). Namun yang dilakukan (Faruk et al., 2018), model Regresi Logistik Biner memiliki kinerja yang baik untuk prediksi, namun kurang baik untuk klasifikasi. Sebaliknya *Random Forest* memiliki kinerja yang sangat baik, untuk prediksi maupun klasifikasi pada data *low birth weight* (LBW). (Kumari & Kr., 2017) juga melakukan review pada klasifikasi biner untuk beberapa *machine learning*.

Metode *oversampling* digunakan untuk menangani *imbalanced data set* dalam kasus algoritma Regresi Logistik Biner. Hasil penelitian ini, metode RWO-*sampling* dengan pendekatan replikasi secara random menunjukkan akurasi yang lebih baik dibandingkan dengan metode RWO-*sampling* dengan pendekatan *roulette* dan ROS. Untuk pengujian masalah underfitting dalam regresi logistik menunjukkan bahwa metode *oversampling* lebih baik daripada *non-oversampling* dengan kenaikan nilai akurasi mencapai rata-rata 2,3% dari setiap data set (Ustyannie & Suprapto, 2020).

Oversampling dan undersampling juga digunakan pada kasus *imbalanced data set* untuk mengklasifikasi kemungkinan penyakit stroke. Hasilnya, teknik *oversampling* mempunyai kinerja lebih tinggi yaitu 95% dibandingkan *undersampling* dengan kinerja 76% (Mutmainah, 2021). K-Means SMOTE juga digunakan untuk menangani *imbalanced data set* dalam klasifikasi penyakit diabetes. Kombinasi K-Means SMOTE dengan metode klasifikasi SVM memiliki akurasi dan sensitifitas terbaik, yaitu sebesar 82% dan 77%, sedangkan dengan metode Naïve Bayes menghasilkan spesifisitas terbaik sebesar 89% (Hairani et al., 2020).

Perbandingan metode SMOTE-N, SMOTE-N-ENN, dan ADASYN-N juga dilakukan dalam menangani imbalanced data set untuk pengklasifikasian seks pranikah pada remaja. Metode terbaik adalah ADASYN-N dengan AUC tertinggi (Fithriasari et al., 2020). Demikian juga dalam (Hayaty et al., 2021), metode SMOTE-*Oversampling* (SOS)

dan *Random Oversampling* (ROS) berhasil meningkatkan akurasi pengenalan kelas yang tidak sesuai dari 12% hingga 100% di algoritma KNN. Sebaliknya, algoritma Naïve Bayes tidak mengalami peningkatan sebelum dan sesudah proses *balancing*, yaitu adalah 89%.

Oversampling dan *Undersampling*

Dalam (Widodo & Handoyo, 2017), *oversampling* merupakan metode pembangkitan data minoritas sebanyak data mayoritas. Pada penelitian ini metode SMOTE diterapkan sebagai teknik *oversampling*. Metode SMOTE berfungsi untuk membuat data buatan, yaitu suatu data replika atau data sintetik dari data minoritas. Berkebalikan dengan *oversampling*, *undersampling* adalah metode untuk mengambil beberapa data mayoritas sehingga jumlah data mayoritas sama besar jumlahnya dengan jumlah data minoritas.

Pengukuran Kinerja Klasifikasi

Percobaan dari penelitian dapat dilakukan sebuah evaluasi dengan pengukuran nilai *accuracy*, *precision*, *recall* dan *f-score*. Menurut Xhemali dalam (Fibrianda & Bhawiyuga, 2018), *Confusion Matrix* dapat dilakukan pengukuran dengan cara menggunakan tabel klasifikasi yang bersifat prediktif (Gambar 3). *Confusion matrix* menurut Han dan Kamber dalam (Fibrianda & Bhawiyuga, 2018) dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.

		Predicted class		
		yes	no	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P′	N′	P + N

Gambar 4. Confusion Matrix menampilkan total positive dan negative tuple

TP (*True-Positive*) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif

FP (*False-Positive*) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif

FN (*False-Negative*) → Jumlah data dengan nilai sebenarnya positif dan nilai prediksi negatif

TN (*True-Negative*) → Jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif

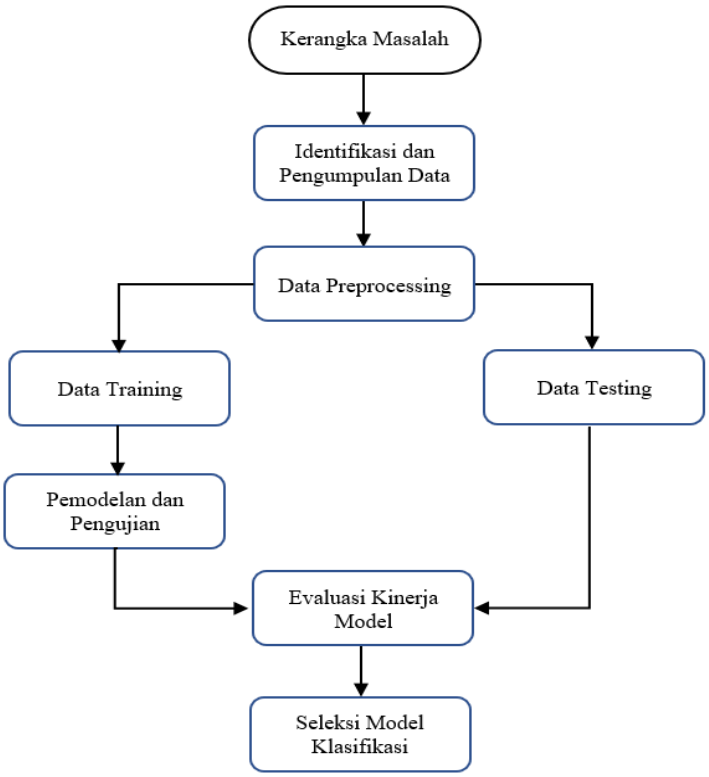
3.4 Metodologi Penelitian

Sumber Data

Data yang digunakan dalam pemodelan klasifikasi adalah data pendaftar Seleksi Bersama Mahasiswa Politeknik Negeri (SBMPN) tahun 2021 yang memilih Politeknik Elektronika Negeri Surabaya (PENS) sebagai pilihan pertamanya.

Metodologi

Metodologi pemodelan regresi logistik dalam penelitian ini dapat dilihat pada Gambar 5, mulai dari penetapan kerangka masalah, pengumpulan data, *data preprocessing*, membagi data *training* dan data *testing*, pemodelan dan pengujian, evaluasi kinerja model, dan seleksi model klasifikasi.



Gambar 6. Metodologi Pemodelan Klasifikasi

Pada tahap *data preprocessing*, total data set asli sebanyak 2556 record dengan 34 fitur. Setelah dilakukan pembersihan data, ada sebanyak 1707 record dan dipilih 8 fitur sebagai variabel prediktor dan variabel respon. *Encoding* dilakukan untuk fitur-fitur dengan tipe data teks menjadi data integer. Dalam model alternatif, dilakukan normalisasi data kemudian membagi data set menjadi data *training* dan data *testing*. Selain itu, pada model alternatif juga dilakukan *undersampling*, *oversampling*, dan kombinasi *oversampling* dan

undersampling. Rasio data *training* dan data *testing* adalah 4:1 untuk semua model yang dicobakan. Total data set beberapa model dapat dilihat pada Tabel 1.

Tabel 1. Data set (Data *Training* dan Data *Testing*)

Model	Data Training	Data Testing	Total Data set
Data Asli	1365	342	1707
Data Hasil Normalisasi	1365	342	1707
Undersampling	417	104	521
Oversampling	2251	563	2814
Oversampling & Undersampling	1366	340	1706

Pada tahap pemodelan dan pengujian, dilakukan pengujian secara statistik signifikansi parameter variabel prediktor dalam model regresi logistik. Sedangkan pada tahap evaluasi kinerja model, dibandingkan nilai-nilai *accuracy*, *precision*, *recall*, F1 *score* atau juga disebut sebagai *F-Measure*, dan *Area Under the Curve* (AUC).

Variabel Penelitian

Beberapa fitur sebagai variabel penelitian yang digunakan dalam pemodelan klasifikasi biner dapat dilihat pada Tabel 2. Untuk menyelesaikan penelitian ini menggunakan pemrograman R, *Python Jupyter Notebook*, dan Microsoft Excel.

Tabel 2. Variabel Penelitian

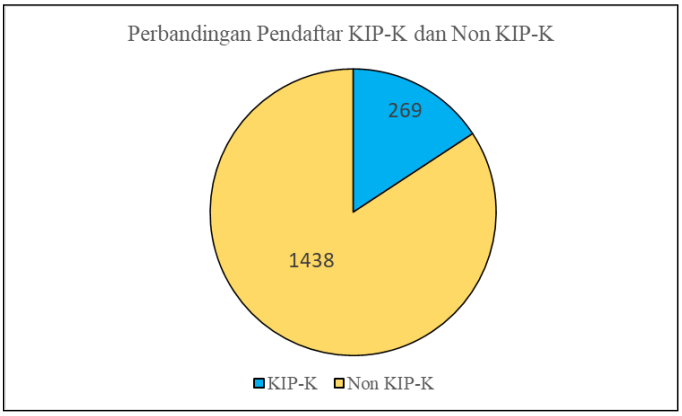
Variabel	**Tipe Variabel [Skala Pengukuran]**	**Keterangan**
Y (Respon)	boolean [nominal]	status pendaftar: 0 : Tidak memilih jalur Kartu Indonesia Pintar – Kuliah (KIP Kuliah) atau Non KIP Kuliah 1 : Memilih jalur KIP Kuliah
Variabel Prediktor		
X_1	integer [nominal]	status orangtua 1 : Ayah dan Ibu masih hidup 2 : Ayah meninggal 3 : Ibu Meninggal 4 : Ayah dan Ibu Meninggal
X2	float [rasio]	penghasilan orangtua
X3	integer [ordinal]	status rumah 0 : kontrak/sewa/menumpang

Variabel	Tipe Variabel [Skala Pengukuran]	Keterangan
		1 : milik sendiri
X4	integer [rasio]	jumlah rumah
X5	integer [rasio]	jumlah motor
X6	integer [rasio]	jumlah mobil
X7	integer [rasio]	daya listrik rumah 1 : 450 Watt 2 : 900 Watt 3 : 1300 Watt atau lebih

3.5 Hasil dan Pembahasan

Statistik Deskriptif

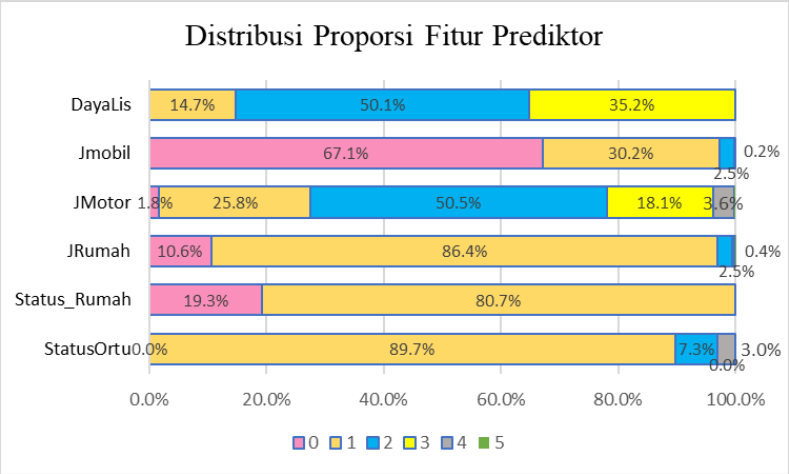
Setelah dilakukan pembersihan data, total data asli yang digunakan sebanyak 1707 record dengan komposisi pendaftar yang mengusulkan skema KIP Kuliah sebanyak 269 record (15,8%) dan pendaftar yang tidak mengusulkan KIP Kuliah sebanyak 1438 record (84,2%), dapat dilihat pada Gambar 7.



Gambar 8. Perbandingan data kelas KIP Kuliah dan Non KIP Kuliah

Gambar 9 adalah distribusi proporsi variabel status orangtua, status rumah, jumlah rumah, jumlah motor, jumlah mobil, dan daya listrik. Berdasarkan status orangtua, ayah dan ibunya masih hidup sebanyak 1531 orang (89,7%), ayah sudah meninggal sebanyak 124 orang (7,3%), dan ayah ibunya sudah meninggal sebanyak 52 orang (3%). Berdasarkan status rumah, kontrak/sewa/ menumpang

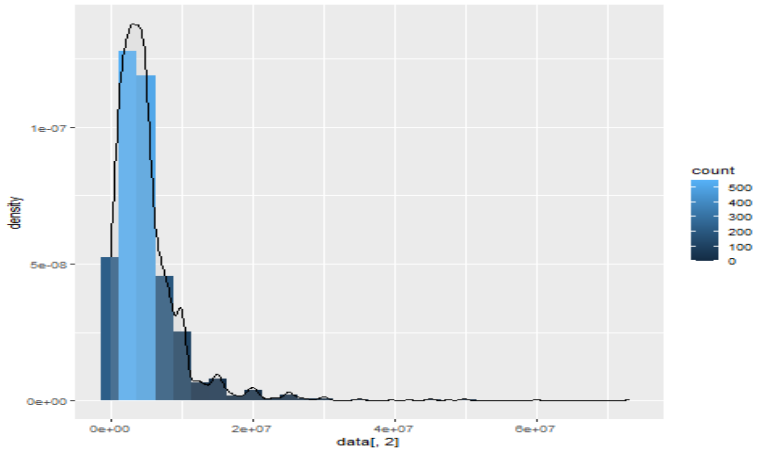
sebanyak 329 orang (19,3%) dan rumah sendiri sebanyak 1378 orang (80,7%).



Gambar 10. Distribusi Proporsi Fitur Prediktor

Berdasarkan jumlah rumah, sebagian besar mempunyai satu rumah, yaitu 86,4%, dan ada sekitar 3,1% yang mempunyai rumah dua atau lebih. Untuk jumlah motor, sebagian besar mempunyai dua motor, yaitu sebanyak 50,5%. Dari 1707 orang, ada 32,9% yang mempunyai mobil. Sedangkan daya listrik, mayoritas pada daya 900 Watt, yaitu sebanyak 50,1%, sedangkan yang lainnya berlanggan daya 450 Watt dan 1300 Watt masing-masing sebanyak 14,7% dan 35,2%.

Pada Gambar 11 distribusi variabel penghasilan adalah *positively skewness* dengan rerata penghasilan sebesar Rp. 5.195.012 dan simpangan baku sebesar Rp. 5.552.922. Nilai *skewness* 4,41 dan kurtosis 32,21. Ini mengindikasikan adanya pencilan (*outlier*), dibuktikan adanya penghasilan sebesar Rp. 73.000.000.



Gambar 12. Detail Histogram Fitur Penghasilan

Beberapa Model Regresi Logistik

Model-1 adalah model regresi logistik dengan data asli hasil *encoding* dengan tujuh variabel prediktor, yaitu status orangtua (X_1), penghasilan (X_2), status rumah (X_3), jumlah rumah (X_4), jumlah motor (X_5), jumlah mobil (X_6), dan daya listrik (X_7). Hasil pengujian parameter masing-masing variabel ditunjukkan pada Tabel 3. Dengan tingkat kepercayaan 95%, ada tiga parameter variabel prediktor yang tidak signifikan dalam model, yaitu status orangtua (X_1) dengan *p-value* sebesar 0,45864, jumlah rumah (X_4) dengan *p-value* sebesar 0,69316, dan daya listrik (X_7) dengan p-*value* sebesar 0,10675.

Tabel 3. Model-1 (Data Asli)

Variabel	Estimasi	Std. Error	z value	Pr(>|z|)
Intercept	1.570e+00	3.894e-01	4.031	5.55e-05 ***
X_1	8.879e-02	1.198e-01	0.741	0.45864
X_2	-4.504e-07	5.369e-08	-8.388	< 2e-16 ***
X_3	-9.510e-01	2.269e-01	-4.192	2.76e-05 ***
X_4	9.279e-02	2.352e-01	0.395	0.69316
X_5	-3.460e-01	1.158e-01	-2.987	0.00282 **
X_6	-1.544e+00	3.606e-01	-4.281	1.86e-05 ***
X_7	-2.166e-01	1.343e-01	-1.613	0.10675

Berikutnya dilakukan pemodelan ulang (Model-2) dengan menghapus tiga variabel yang tidak signifikan. Hasil pengujian parameter masing-masing variabel ditunjukkan pada Tabel 4. Dengan tingkat kepercayaan 95%, parameter *intercept*, penghasilan (X_2), status rumah (X_3), jumlah motor (X_5), dan jumlah mobil (X_6) tetap signifikan dalam model dengan *p-value* masing-masing sebesar 3,62e-08, 2e-16, 3,70e-06, 0,00195, dan 6,32e-06.

Tabel 4. Model-2 (Data Asli)

Variabel	Estimasi	Std. Error	z value	Pr(>|z|)
Intercept	1.352e+00	2.455e-01	5.508	3.62e-08 ***
X_2	-4.745e-07	5.204e-08	-9.118	< 2e-16 ***
X_3	-8.550e-01	1.848e-01	-4.628	3.70e-06 ***
X_5	-3.574e-01	1.154e-01	-3.097	0.00195 **
X_6	-1.612e+00	3.571e-01	-4.515	6.32e-06 ***

Pemodelan regresi logistik data hasil normalisasi semua variabel prediktor sebagai Model-3. Dengan tingkat kepercayaan 95%, ada tiga parameter variabel prediktor yang tidak signifikan dalam model, yaitu status orangtua (X_1) dengan *p-value* sebesar 0,45864, jumlah rumah

(X_4) dengan *p-value* sebesar 0,69316, dan daya listrik (X_7) dengan *p-value* sebesar 0,10675. Sedangkan Model-4 adalah model regresi logistik dengan data hasil normalisasi dengan menghapus tiga variabel yang tidak signifikan. Dengan tingkat kepercayaan 95%, didapatkan hasil yang sama dengan Model-2, yang mana parameter *intercept*, penghasilan (X_2), status rumah (X_3), jumlah motor (X_5), dan jumlah mobil (X_6) signifikan dalam model dengan *p-value* masing-masing sebesar 2e-16, 2e-16, 3,70e-06, 0,00195, dan 6,32e-06.

Berikutnya dilakukan pemodelan dengan melakukan *undersampling* sebagai Model-5. Pada model tersebut, hanya memasukkan empat variabel prediktor yang signifikan dalam model-model sebelumnya. Hasil pengujian parameter masing-masing variabel ditunjukkan pada Tabel 5.

Tabel 5. Model-5 (*Undersampling* Data Asli)

Variabel	Estimasi	Std. Error	z value	Pr(>|z|)
Intercept	3.680e+00	4.431e-01	8.305	< 2e-16 ***
X_2	-4.807e-07	7.051e-08	-6.818	9.22e-12 ***
X_3	-1.167e+00	3.029e-01	-3.854	0.000116 ***
X_5	-5.814e-01	1.688e-01	-3.445	0.000570 ***
X_6	-1.055e+00	4.285e-01	-2.462	0.013823 *

Dengan tingkat kepercayaan 95%, parameter *intercept* signifikan dalam model dengan *p-value* sebesar 2e-16, penghasilan dengan *p-value* sebesar 9,22e-12, status rumah dengan p-value sebesar 0,000116, jumlah motor dengan *p-value* sebesar 0,000570, dan jumlah mobil dengan p-value sebesar 0,013823. Model-6 adalah pemodelan dengan melakukan *oversampling*. Hasil pengujian parameter masing-masing variabel ditunjukkan pada Tabel 6.

Tabel 6. Model-6 (*Oversampling* Data Asli)

Variabel	Estimasi	Std. Error	z value	Pr(>|z|)
Intercept	3.082e+00	1.767e-01	17.445	< 2e-16 ***
X_2	-4.800e-07	3.090e-08	-15.535	< 2e-16 ***
X_3	-9.526e-01	1.232e-01	-7.731	1.06e-14 ***
X_5	-3.834e-01	7.254e-02	-5.286	1.25e-07 ***
X_6	-1.223e+00	1.677e-01	-7.294	3.02e-13 ***

Dengan tingkat kepercayaan 95%, parameter *intercept* signifikan dalam model dengan *p-value* sebesar 2e-16, penghasilan dengan *p-value* sebesar 2e-16, status rumah dengan p-value sebesar 1,06e-14, jumlah motor dengan *p-value* sebesar 1,25e-07, dan jumlah mobil dengan *p-value* sebesar 3,02e-13.

Model-7 adalah pemodelan dengan melakukan kombinasi *oversampling* dan *undersampling*. Dengan tingkat kepercayaan 95%, parameter *intercept* signifikan dalam model dengan p-value sebesar

2e-16, penghasilan dengan *p-value* sebesar 2e-16, status rumah dengan p-value sebesar 4,77e-12, jumlah motor dengan p-*value* sebesar 0,000169, dan jumlah mobil dengan *p-value* sebesar 1,37e-05.

Evaluasi Kinerja Model

Langkah berikutnya adalah mengevaluasi tujuh model yang telah dibahas dengan membandingkan metrik dalam matriks konfusi, yaitu *accuracy*, *precision*, *recall*, dan F1 *Score*. Berdasarkan Tabel 7, nilai *accuracy* Model-2 mempunyai *accuracy* paling tinggi, yaitu sebesar 88,01%. *Accuracy* tertinggi kedua adalah Model-4, yaitu sebesar 86,26%. Berikutnya Model-5 (*undersampling*) dengan *accuracy* sebesar 81,73%. Sedangkan Model-6 (*oversampling*) dan Model-7 (*oversampling* dan *undersampling*) masing-masing *accuracy* sebesar 77,62% dan 76,47%.

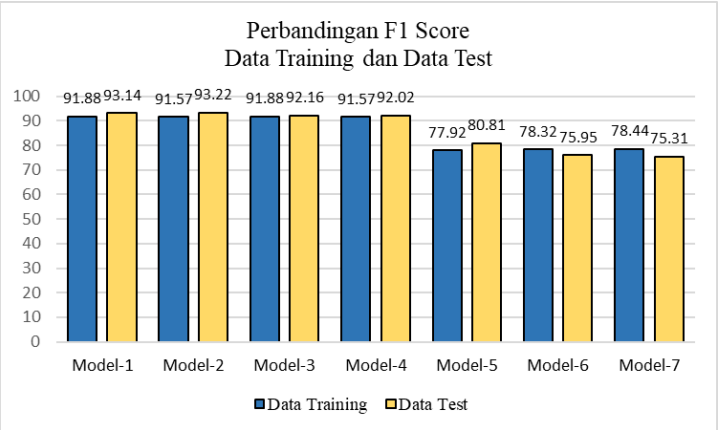
Tabel 7. Perbandingan Kinerja Model

Model	***Accuracy***	***Precision***	***Recall***	**F1 *Score***
Model-1	87.72	98.96	87.96	93.14
Model-2	88.01	97.92	88.96	93.22
Model-3	85.96	97.92	87.04	92.16
Model-4	86.26	94.10	90.03	92.02
Model-5	81.73	80.00	81.63	80.81
Model-6	77.62	69.10	84.32	75.95
Model-7	76.47	69.32	82.43	75.31

Berdasarkan nilai *precision*, Model-2 mempunyai *precision* paling tinggi, yaitu sebesar 97,92%. *Precision* tertinggi kedua adalah Model-4, yaitu sebesar 94,10%. Berikutnya Model-5 (*undersampling*) dengan *precision* sebesar 80,00%. Sedangkan Model-6 (*oversampling*) sebesar 69,10% dan Model-7 (*oversampling* dan *undersampling*) sebesar 69,32%.

Demikian juga untuk nilai *recall*, Model-4 mempunyai *recall* tertinggi, yaitu sebesar 90,03%. Recall tertinggi kedua adalah Model-2, yaitu sebesar 88,96%. Berikutnya Model-5 (*undersampling*) dengan *recall* sebesar 81,63%. Sedangkan Model-6 (*oversampling*) sebesar 84,32% dan Model-7 (*oversampling* dan *undersampling*) sebesar 82,43%.

Dari beberapa model yang dicoba, jumlah *False Positive* (FP) dan *False Negative* cenderung tidak simetri, sehingga F1 *Score* lebih dipertimbangkan daripada nilai *accuracy*. Gambar 13 merupakan perbandingan F1 *Score* data *training* dan data *testing* pada tujuh model yang dicobakan.

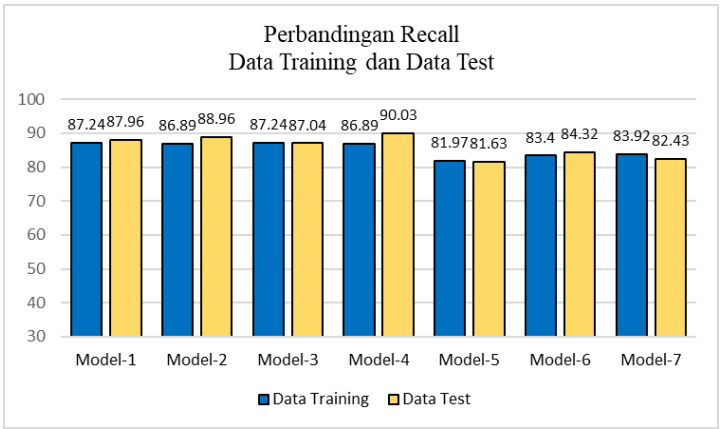


Gambar 14. Perbandingan F1 Score Model-1 sd. Model-7

Berdasarkan gambar tersebut dapat dilihat bahwa Model-2 mempunyai Rerata F1 *Score* tertinggi (92,40%), yaitu F1 *Score* data *training* sebesar 91,57% dan F1 *Score* data *testing* sebesar 93,22%., Rerata F1 *Score* terbesar kedua adalah Model-4 yaitu sebesar 91,80% (F1 *Score* data *training* 91,57% dan F1 *Score* data *testing* 92,02%. Sedangkan rerata F1 *Score* Model-5, Model-6, dan Model-7 masing-masing sebesar 79,37%, 77,14%, dan 76,88%.

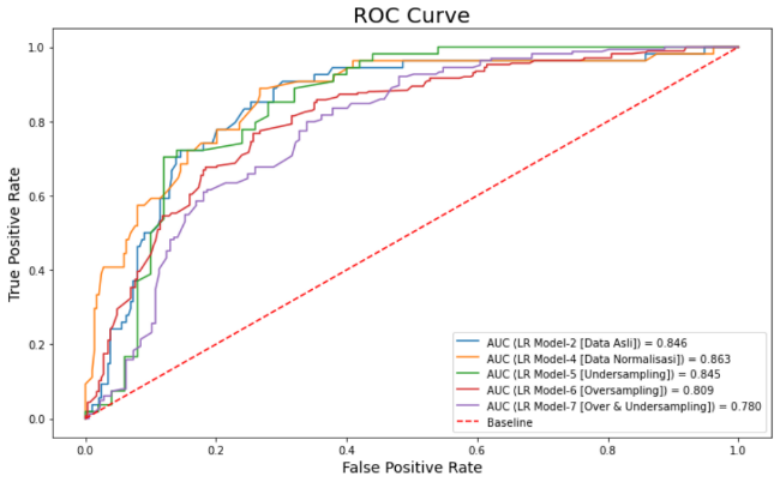
Evaluasi utama lain yang dipertimbangkan untuk model klasifikasi biner ini adalah nilai *recall*. Nilai *recall* lebih dipertimbangkan daripada nilai *precision* karena lebih baik memprediksi mahasiswa memilih KIP Kuliah tetapi sebenarnya tidak memilih KIP Kuliah (Non KIP Kuliah) daripada salah memprediksi, yaitu mahasiswa diprediksi tidak memilih KIP Kuliah padahal sebenarnya memilih KIP Kuliah.

Pada Gambar 15, rerata *recall* tertinggi adalah Model-4 yaitu sebesar 88,46%. Rerata *recall* tertinggi kedua adalah Model-2 yaitu sebesar 87,93%. Rerata tertinggi ketiga yaitu Model-6 dengan *recall* sebesar 83,86%. Dan berikutnya Model-7 dan Model-5 masing-masing nilai *recall* sebesar 83,18% dan 81,80%.



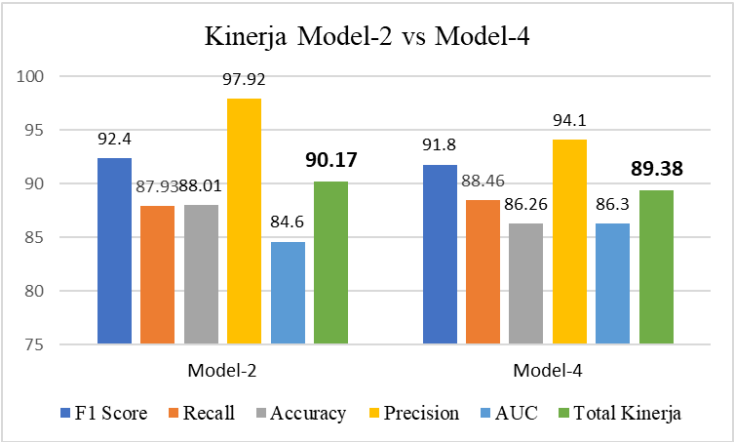
Gambar 16. Perbandingan Recall Model-1

Gambar 17 adalah kurva *Receiver Operating Characteristics* (ROC) dari lima model dengan semua parameter variabel prediktor signifikan.



Gambar 18. Kurva ROC dari 5 model dengan parameter signifikan

Berdasarkan beberapa evaluasi tersebut kinerja Model-2 hampir sama dengan kinerja Model-4. Gambar 19 adalah perbandingan beberapa metrik dari matriks konfusi Model-2 dan Model-4. Jika kinerja dihitung berdasarkan rerata dari nilai F1 *Score*, *recall*, *accuracy*, *precision*, dan AUC maka dipilih Model-2 dengan rerata F1 *Score* 92,40%, rerata *recall* sebesar 87,93%, *accuracy* sebesar 88,01%, *precision* sebesar 97,92%, dan AUC sebesar 84,6%.



Gambar 20. Perbandingan Kinerja Model-2 dan Model-4

Sedangkan rerata kinerja sebesar secara keseluruhan sebesar 90,17%, lebih tinggi 0,79% jika dibandingkan dengan Model-4 (89,38%). Berdasarkan estimasi parameter Model-2 pada Tabel 6, model Regresi Logistik Biner yang direkomendasikan untuk digunakan seleksi mahasiswa baru penerima KIP Kuliah adalah Model-2 dengan fungsi sebagai berikut:

$$P(Y) = \frac{e^{1.352 - 0.0000004745X_2 - 0.855X_3 - 0.357X_5 - 1.612X_6}}{1 + e^{1.352 - 0.0000004745X_2 - 0.855X_3 - 0.357X_5 - 1.612X_6}}$$

Jika P(Y) ≥ 0,5 maka diklasifikasikan sebagai Penerima KIP Kuliah, sedangkan jika P(Y) < 0,5 maka diklasifikasikan sebagai Bukan Penerima KIP Kuliah.

3.6 Kesimpulan

Dari hasil penelitian ini, model terbaik untuk kasus ini adalah Model-2 yaitu model dengan data set asli yang mempunyai rerata F1 *Score* 92,40% dan rerata *recall* sebesar 87,93%, *accuracy* sebesar 88,01%, *precision* sebesar 97,92%, dan AUC sebesar 84,6%. Beberapa fitur atau variabel prediktor yang signifikan dalam model tersebut adalah penghasilan, status rumah, jumlah motor, dan jumlah mobil.

Model klasifikasi ini dapat digunakan sebagai model atau acuan seleksi mahasiswa baru penerima KIP Kuliah dengan membenamkan fungsi model terbaik ke dalam aplikasi yang saat ini digunakan. Pemodelan ulang dapat dilakukan kembali ketika fitur-fitur dalam aplikasi seleksi pendaftaran mahasiswa baru berkembang.

Daftar Pustaka

Ambika, P., Laxmi Lydia, E., Shankar, K., Nguyen, P. T., & Abadi, S. (2019). Logistic regression for health profiling. *International Journal of Engineering and Advanced Technology*, *8*(6 Special Issue 2), 974–977. https://doi.org/10.35940/ijeat.F1294.0886S219

Brahmantyo, Y.-, Riaman, R., & Sukono, F. (2021). Willingness to Pay of Fishermen Insurance Using Logistic Regression with Parameter Estimated by Maximum Likelihood Estimation Based on Newton Raphson Iteration. *Jurnal Matematika Integratif*, *17*(1), 15. https://doi.org/10.24198/jmi.v17.n1.32037.15-21

Chairunnisa, Nasution, Y. N., & Purnamasari, I. (2017). Penerapan Metode Analisis Regresi Logistik Biner Dan Classification And Regression Tree (CART) Pada Faktor yang Mempengaruhi Lama Masa Studi Mahasiswa. *Eksponensial*, *8*(2), 125–134.

Esquivel, J. A., & Esquivel, J. A. (2020). Using a Binary Classification Model to Predict the Likelihood of Enrolment to the Undergraduate Program of a Philippine University. *International Journal of Computer Trends and Technology*, *68*(5), 6–10. https://doi.org/10.14445/22312803/ijctt-v68i5p103

Faruk, A., Cahyono, E. S., Eliyati, N., & Arifieni, I. (2018). Prediction and classification of low birth weight data using machine learning techniques. *Indonesian Journal of Science and Technology*, *3*(1), 18–28. https://doi.org/10.17509/ijost.v3i1.10799

Fibrianda, M. F., & Bhawiyuga, A. (2018). Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *II*(9), 3112–3123.

Fithriasari, K., Hariastuti, I., & Wening, K. S. (2020). Handling Imbalance Data in Classification Model with Nominal Predictors. *International Journal of Computing Science and Applied Mathematics*, *6*(1), 33. https://doi.org/10.12962/j24775401.v6i1.6643

Hairani, H., Saputro, K. E., & Fadli, S. (2020). K-means-SMOTE for handling class imbalance in the classification of diabetes with C4.5, SVM, and naive Bayes. *Jurnal Teknologi Dan Sistem Komputer*, *8*(2), 89–93. https://doi.org/10.14710/jtsiskom.8.2.2020.89-93

Hayaty, M., Muthmainah, S., & Ghufran, S. M. (2021). Random and Synthetic Over-Sampling Approach to Resolve Data Imbalance in Classification. *International Journal of Artificial Intelligence Research*, *4*(2), 86. https://doi.org/10.29099/ijair.v4i2.152

Kemendikbud. (2021). Pedoman Pendaftaran Kartu Indonesia Pintar Kuliah (KIP-K). In *Puslapdik* (Vol. 1, Issue 1). https://kip-kuliah.kemdikbud.go.id/uploads/BsImnu09yFOxop5dfJAwkaRleMTUqP_tgl20200412205459.pdf

Khoirunissa, H. A., Widyaningrum, A. R., & Maharani, A. P. A. (2021). Comparison of Random Forest, Logistic Regression, and MultilayerPerceptron Methods on Classification of Bank Customer Account Closure. In *Indonesian Journal of Applied Statistics* (Vol. 4, Issue 1). https://doi.org/10.13057/ijas.v4i1.41461

Kumari, R., & Kr., S. (2017). Machine Learning: A Review on Binary Classification. *International Journal of Computer Applications*, *160*(7), 11–15. https://doi.org/10.5120/ijca2017913083

Misna, Rais, & Utami, I. T. (2018). Analisis Regresi Logistik Biner Untuk Mengklasifikasi Penderita Hipertensi Berdasarkan Kebiasaan Merokok Di RSU Mokopido Toli-Toli. *Journal of Science and Technology*, *7*(3), 341–348.

Mutmainah, S. (2021). Penanganan Imbalance Data Pada Klasifikasi Kemungkinan Penyakit Stroke. *Jurnal SNATi*, *1*, 10–16. https://library.uii.ac.id/osr

Salamah, U., & Ramayanti, D. (2018). Implementation of Logistic Regression Algorithm for Complaint Text Classification in Indonesian Ministry of Marine and Fisheries Abstract : *International Journal of Computer Techniques*, *5*(5), 74–78. http://www.ijctjournal.org

Shobri, M. Q., Yanuar, F., & Devianto, D. (2021). Covid-19 Patient Mortality Risk Classification Using Bayesian Binary Logistic Regression. *Jurnal*

Matematika, Statistika Dan Komputasi, *18*(1), 150–160. https://doi.org/10.20956/j.v18i1.14268

Smita, M. (2021). Logistic Regression Model For Predicting Performance of S&P BSE30 Company Using IBM SPSS. *International Journal of Mathematics Trends and Technology*, *67*(7), 118–134. https://doi.org/10.14445/22315373/ijmtt-v67i7p515

Suniantara, I. K. P., Suwardika, G., & Astapa, I. G. A. (2018). Bagging Regresi Logistik Pada Peningkatan Ketepatan Klasifikasi Waktu Kelulusan Mahasiswa Stikom Bali. *Dinamika*.

Tampil, Y., Komaliq, H., & Langi, Y. (2017). Analisis Regresi Logistik Untuk Menentukan Faktor-Faktor Yang Mempengaruhi Indeks Prestasi Kumulatif (IPK) Mahasiswa FMIPA Universitas Sam Ratulangi Manado. In *d'CARTESIAN* (Vol. 6, Issue 2). https://doi.org/10.35799/dc.6.2.2017.17023

Ustyannie, W., & Suprapto, S. (2020). Oversampling Method To Handling Imbalanced Data sets Problem in Binary Logistic Regression Algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *14*(1), 1. https://doi.org/10.22146/ijccs.37415

Widodo, A., & Handoyo, S. (2017). The classification performance using logistic regression and support vector machine (SVM). *Journal of Theoretical and Applied Information Technology*, *95*(19), 5184–5193. www.jatit.org

LAMPIRAN

```
import pandas as pd
import numpy as np
from sklearn.datasets import make_classification
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from collections import Counter
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import seaborn as sn
import matplotlib.pyplot as plt
```

```
dataset = pd.read_csv('c:/tugas_sdt/klasifikasi__UKT_komplit.csv')
```

```
dataset.head(10)
```

	No.	StatusOrtu	Penghasilan	Status_Rumah	JMotor	Jmobil	DayaLis	KIPK
0	1	1	4000000	1	1	0	2	0
1	2	1	2500000	0	1	0	3	0
2	3	1	6000000	1	2	0	2	0
3	4	1	5440500	1	2	0	2	0
4	5	1	10000000	0	1	1	3	0
5	6	1	1000000	0	1	0	3	1
6	7	1	20000000	1	2	1	3	0
7	8	1	15000000	1	1	0	3	0
8	9	4	4000000	1	1	1	3	0
9	10	1	0	1	2	0	1	0

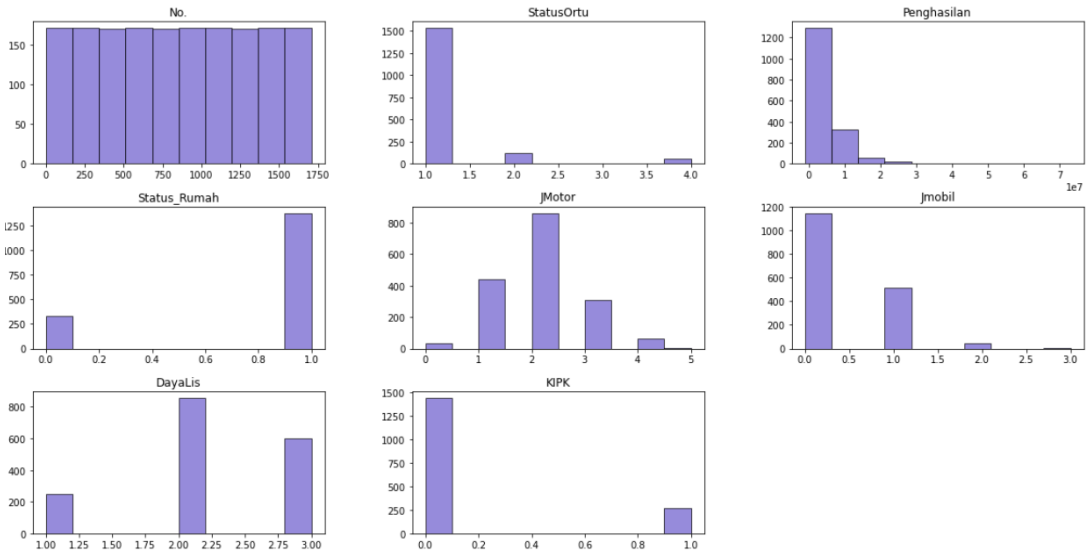
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1707 entries, 0 to 1706
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   No.           1707 non-null   int64
 1   StatusOrtu    1707 non-null   int64
 2   Penghasilan   1707 non-null   int64
 3   Status_Rumah  1707 non-null   int64
 4   JMotor        1707 non-null   int64
 5   Jmobil        1707 non-null   int64
 6   DayaLis       1707 non-null   int64
 7   KIPK          1707 non-null   int64
dtypes: int64(8)
memory usage: 106.8 KB
```

```
dataset.describe()
```

	No.	StatusOrtu	Penghasilan	Status_Rumah	JMotor	Jmobil	DayaLis	KIPK
count	1707.000000	1707.000000	1.707000e+03	1707.000000	1707.000000	1707.000000	1707.000000	1707.000000
mean	854.000000	1.164030	5.195012e+06	0.807264	1.968366	0.357938	2.205038	0.157586
std	492.912771	0.565764	5.552922e+06	0.394563	0.823274	0.541534	0.676276	0.364460
min	1.000000	1.000000	-1.000000e+06	0.000000	0.000000	0.000000	1.000000	0.000000
25%	427.500000	1.000000	2.000000e+06	1.000000	1.000000	0.000000	2.000000	0.000000
50%	854.000000	1.000000	4.000000e+06	1.000000	2.000000	0.000000	2.000000	0.000000
75%	1280.500000	1.000000	6.131916e+06	1.000000	2.000000	1.000000	3.000000	0.000000
max	1707.000000	4.000000	7.300000e+07	1.000000	5.000000	3.000000	3.000000	1.000000

```
user_data_for_hist = \
dataset
user_data_for_hist.hist(figsize=(20,10),alpha = 0.7, color = 'slateblue',edgecolor ='black',grid=False)
```




```
from sklearn.preprocessing import MinMaxScaler
# variabel prediktor (input)
#X = dataset.iloc[:, [3,4]].values
# klasifikasi (output)
#y = dataset.iloc[:,-1].values
array =dataset.values
X = array[:,1:7] #slicing dataframe ke dalam array
y = array[:,7]

scaler = MinMaxScaler()
# transform data
X = scaler.fit_transform(X)
```

```
y= y.astype('int')
print(y)
```

```
[0 0 0 ... 0 0 0]
```

```
print(Counter(y))
```

```
Counter({0: 1438, 1: 269})
```

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)
```

```
# instantiating the random over sampler_
ros = RandomOverSampler()
# resampling x_train, y_train
X_ros, y_ros = ros.fit_resample(x_train, y_train)
# new class distribution_
print(Counter(y_ros))
```

```
Counter({0: 1156, 1: 1156})
```

```
# instantiating the random under sampler_
rus = RandomUnderSampler()
# resampling x_train, y_train
X_rus, y_rus = rus.fit_resample(x_train, y_train)
# new class distribution_
print(Counter(y_rus))
```

```
Counter({0: 209, 1: 209})
```

```
# instantiating over and under sampler
over = RandomOverSampler(sampling_strategy=0.5)
under = RandomUnderSampler(sampling_strategy=0.8)
# first performing oversampling to minority class
X_over, y_over = over.fit_resample(x_train, y_train)
print(f"Oversampled: {Counter(y_over)}")
```

```
Oversampled: Counter({0: 1156, 1: 578})
```

```
# now to combine under sampling
X_comb, y_comb = under.fit_resample(X_over, y_over)
print(f"Combined Random Sampling: {Counter(y_comb)}")
```

```
Combined Random Sampling: Counter({0: 722, 1: 578})
```

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
X_smote, y_smote = sm.fit_resample(x_train, y_train)
print(f"SMOTE Random Sampling: {Counter(y_smote)}")
```

```
SMOTE Random Sampling: Counter({0: 1156, 1: 1156})
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
```

```
#from sklearn.svm import SVM
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix,accuracy_score

logmodel = LogisticRegression()

#1. Model1 dengan dataset asli
model1=logmodel.fit(x_train, y_train)
predictions1a = model1.predict(x_train)
predictions1b = model1.predict(x_test)
predictions1c = model1.predict_proba(x_test)[:,1]
print("-------Model-1: Logit Biner dengan Dataset Asli-------")
print("Kinerja Data Training:")
print(classification_report(y_train, predictions1a))
print(confusion_matrix(y_train, predictions1a))
print(accuracy_score(y_train, predictions1a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions1b))
print(confusion_matrix(y_test, predictions1b))
print(accuracy_score(y_test, predictions1b))


#2. Model2 dengan dataset over-sampling
model2=logmodel.fit(X_ros, y_ros)
predictions2a = model2.predict(X_ros)
predictions2b = model2.predict(x_test)
predictions2c = model2.predict_proba(x_test)[:,1]
print("-------Model-2: Logit Biner dengan Dataset Over-sampling---
----")
print("Kinerja Data Training:")
print(classification_report(y_ros, predictions2a))
print(confusion_matrix(y_ros, predictions2a))
print(accuracy_score(y_ros, predictions2a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions2b))
print(confusion_matrix(y_test, predictions2b))
print(accuracy_score(y_test, predictions2b))

#3. Model3 dengan dataset under-sampling
model3=logmodel.fit(X_rus, y_rus)
predictions3a = model3.predict(X_rus)
predictions3b = model3.predict(x_test)
predictions3c = model3.predict_proba(x_test)[:,1]
print("-------Model-3: Logit Biner dengan Dataset Under-sampling--
-----")
print("Kinerja Data Training:")
print(classification_report(y_rus, predictions3a))
print(confusion_matrix(y_rus, predictions3a))
print(accuracy_score(y_rus, predictions3a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions3b))
print(confusion_matrix(y_test, predictions3b))
print(accuracy_score(y_test, predictions3b))

#4. Model4 dengan dataset kombinasi
model4=logmodel.fit(X_comb, y_comb)
predictions4a = model4.predict(X_comb)
```

```
predictions4b = model4.predict(x_test)
predictions4c = model4.predict_proba(x_test)[:,1]
print("-------Model-4: Logit Biner dengan Dataset Kombinasi Over-
Under-------")
print("Kinerja Data Training:")
print(classification_report(y_comb, predictions4a))
print(confusion_matrix(y_comb, predictions4a))
print(accuracy_score(y_comb, predictions4a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions4b))
print(confusion_matrix(y_test, predictions4b))
print(accuracy_score(y_test, predictions4b))

#5. Model5 dengan dataset SMOTE
model5=logmodel.fit(X_smote, y_smote)
predictions5a = model5.predict(X_smote)
predictions5b = model5.predict(x_test)
predictions5c = model5.predict_proba(x_test)[:,1]
print("-------Model-5: Logit Biner dengan Dataset SMOTE-------")
print("Kinerja Data Training:")
print(classification_report(y_smote, predictions5a))
print(confusion_matrix(y_smote, predictions5a))
print(accuracy_score(y_smote, predictions5a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions5b))
print(confusion_matrix(y_test, predictions5b))
print(accuracy_score(y_test, predictions5b))

#6. Model6 dengan dataset SMOTE dan metode Random Forest
modelRF = RandomForestClassifier()
model6=modelRF.fit(x_train, y_train)
predictions6a = model6.predict(x_train)
predictions6b = model6.predict(x_test)
predictions6c = model6.predict_proba(x_test)[:,1]
print("-------Model-6: Random Forest dengan Dataset Asli-------")
print("Kinerja Data Training:")
print(classification_report(y_train, predictions6a))
print(confusion_matrix(y_train, predictions6a))
print(accuracy_score(y_train, predictions6a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions6b))
print(confusion_matrix(y_test, predictions6b))
print(accuracy_score(y_test, predictions6b))

#7. Model7 dengan dataset asli dan metode Random Forest
modelRF = RandomForestClassifier()
model7=modelRF.fit(X_smote, y_smote)
predictions7a = model7.predict(X_smote)
predictions7b = model7.predict(x_test)
predictions7c = model7.predict_proba(x_test)[:,1]
print("-------Model-7: Random Forest dengan Dataset SMOTE-------")
print("Kinerja Data Training:")
print(classification_report(y_smote, predictions7a))
print(confusion_matrix(y_smote, predictions7a))
print(accuracy_score(y_smote, predictions7a))
print("Kinerja Data Testing:")
print(classification_report(y_test, predictions7b))
```

```
print(confusion_matrix(y_test, predictions7b))
print(accuracy_score(y_test, predictions7b))
```

```
-------Model-1: Logit Biner dengan Dataset Asli-------
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.85      0.99      0.92      1156
           1       0.52      0.07      0.13       209

    accuracy                           0.85      1365
   macro avg       0.69      0.53      0.52      1365
weighted avg       0.80      0.85      0.80      1365

[[1142   14]
 [ 194   15]]
0.8476190476190476
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.83      0.98      0.90       282
           1       0.30      0.05      0.09        60

    accuracy                           0.81       342
   macro avg       0.56      0.51      0.49       342
weighted avg       0.74      0.81      0.75       342

[[275   7]
 [ 57   3]]
0.8128654970760234
-------Model-2: Logit Biner dengan Dataset Over-sampling-------
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.80      0.71      0.75      1156
           1       0.74      0.83      0.78      1156

    accuracy                           0.77      2312
   macro avg       0.77      0.77      0.77      2312
weighted avg       0.77      0.77      0.77      2312

[[815 341]
 [199 957]]
0.7664359861591695
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.96      0.76      0.85       282
           1       0.43      0.85      0.57        60

    accuracy                           0.77       342
   macro avg       0.69      0.80      0.71       342
weighted avg       0.87      0.77      0.80       342

[[214  68]
 [  9  51]]
0.7748538011695907
```

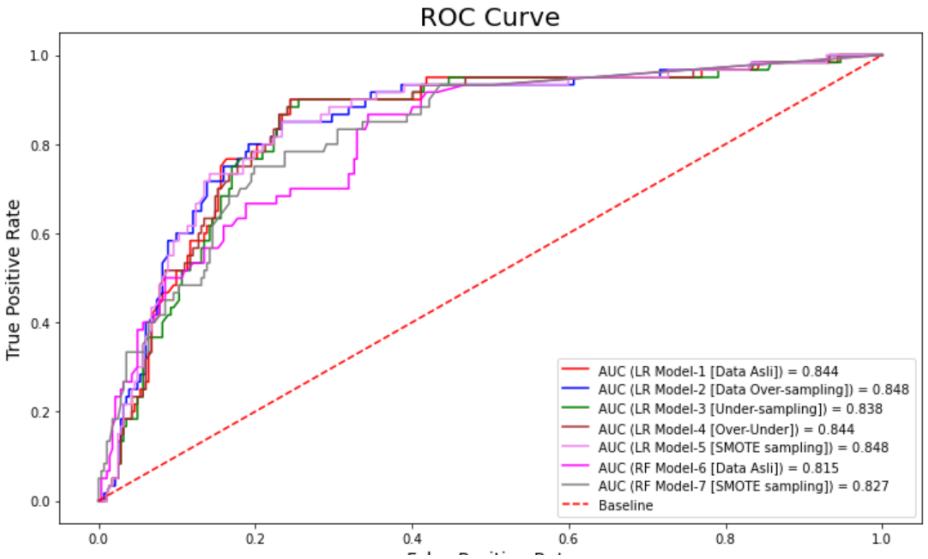
```
-------Model-3: Logit Biner dengan Dataset Under-sampling-------
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.74      0.74      0.74       209
           1       0.74      0.74      0.74       209

    accuracy                           0.74       418
   macro avg       0.74      0.74      0.74       418
weighted avg       0.74      0.74      0.74       418

[[154  55]
 [ 55 154]]
0.7368421052631579
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.96      0.77      0.85       282
           1       0.44      0.87      0.58        60

    accuracy                           0.78       342
   macro avg       0.70      0.82      0.72       342
weighted avg       0.87      0.78      0.81       342

[[216  66]
 [  8  52]]
0.783625730994152
-------Model-4: Logit Biner dengan Dataset Kombinasi Over-Under-----
--
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.77      0.76      0.76       722
           1       0.70      0.72      0.71       578

    accuracy                           0.74      1300
   macro avg       0.74      0.74      0.74      1300
weighted avg       0.74      0.74      0.74      1300

[[547 175]
 [162 416]]
0.7407692307692307
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.95      0.79      0.86       282
           1       0.44      0.80      0.57        60

    accuracy                           0.79       342
   macro avg       0.70      0.79      0.72       342
weighted avg       0.86      0.79      0.81       342

[[222  60]
 [ 12  48]]
0.7894736842105263
-------Model-5: Logit Biner dengan Dataset SMOTE-------
Kinerja Data Training:
              precision    recall  f1-score   support
```

```
           0       0.81      0.71      0.75      1156
           1       0.74      0.83      0.78      1156

    accuracy                           0.77      2312
   macro avg       0.77      0.77      0.77      2312
weighted avg       0.77      0.77      0.77      2312

[[820 336]
 [198 958]]
0.7690311418685121
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.96      0.77      0.85       282
           1       0.44      0.85      0.58        60

    accuracy                           0.78       342
   macro avg       0.70      0.81      0.71       342
weighted avg       0.87      0.78      0.80       342

[[216  66]
 [  9  51]]
0.7807017543859649
-------Model-6: Random Forest dengan Dataset Asli-------
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96      1156
           1       0.88      0.62      0.73       209

    accuracy                           0.93      1365
   macro avg       0.91      0.80      0.84      1365
weighted avg       0.93      0.93      0.92      1365

[[1138   18]
 [  79  130]]
0.928937728937729
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.86      0.95      0.91       282
           1       0.56      0.30      0.39        60

    accuracy                           0.84       342
   macro avg       0.71      0.63      0.65       342
weighted avg       0.81      0.84      0.82       342

[[268  14]
 [ 42  18]]
0.8362573099415205
-------Model-7: Random Forest dengan Dataset SMOTE-------
Kinerja Data Training:
              precision    recall  f1-score   support

           0       0.96      0.94      0.95      1156
           1       0.94      0.96      0.95      1156
```

```
    accuracy                           0.95      2312
   macro avg       0.95      0.95      0.95      2312
weighted avg       0.95      0.95      0.95      2312

[[1081   75]
 [  49 1107]]
0.9463667820069204
Kinerja Data Testing:
              precision    recall  f1-score   support

           0       0.89      0.87      0.88       282
           1       0.45      0.52      0.48        60

    accuracy                           0.80       342
   macro avg       0.67      0.69      0.68       342
weighted avg       0.82      0.80      0.81       342

[[244  38]
 [ 29  31]]
0.804093567251462
```

```
from sklearn.metrics import roc_auc_score, roc_curve

#y_test_int = y_test.replace({'Good': 1, 'Bad': 0})
auc1 = roc_auc_score(y_test, predictions1c)
fpr1, tpr1, thresholds1 = roc_curve(y_test, predictions1c)

auc2 = roc_auc_score(y_test, predictions2c)
fpr2, tpr2, thresholds2 = roc_curve(y_test, predictions2c)

auc3 = roc_auc_score(y_test, predictions3c)
fpr3, tpr3, thresholds3 = roc_curve(y_test, predictions3c)

auc4 = roc_auc_score(y_test, predictions4c)
fpr4, tpr4, thresholds4 = roc_curve(y_test, predictions4c)

auc5 = roc_auc_score(y_test, predictions5c)
fpr5, tpr5, thresholds5 = roc_curve(y_test, predictions5c)

auc6 = roc_auc_score(y_test, predictions6c)
fpr6, tpr6, thresholds6 = roc_curve(y_test, predictions6c)

auc7 = roc_auc_score(y_test, predictions7c)
fpr7, tpr7, thresholds7 = roc_curve(y_test, predictions7c)
```

```
plt.figure(figsize=(12, 7))
plt.plot(fpr1, tpr1, label=f'AUC (LR Model-1 [Data Asli]) = {auc1:.3f}',color='red')
plt.plot(fpr2, tpr2, label=f'AUC (LR Model-2 [Data Over-sampling]) = {auc2:.3f}',color='blue')
plt.plot(fpr3, tpr3, label=f'AUC (LR Model-3 [Under-sampling]) = {auc3:.3f}',color='green')
plt.plot(fpr4, tpr4, label=f'AUC (LR Model-4 [Over-Under]) = {auc4:.3f}',color='brown')
plt.plot(fpr5, tpr5, label=f'AUC (LR Model-5 [SMOTE sampling]) = {auc5:.3f}',color='violet')
plt.plot(fpr6, tpr6, label=f'AUC (RF Model-6 [Data Asli]) = {auc6:.3f}',color='magenta')
plt.plot(fpr7, tpr7, label=f'AUC (RF Model-7 [SMOTE sampling]) = {auc7:.3f}',color='grey')


plt.plot([0, 1], [0, 1], color='red', linestyle='--', label='Baseline')
plt.title('ROC Curve', size=20)
plt.xlabel('False Positive Rate', size=14)
plt.ylabel('True Positive Rate', size=14)
plt.legend();
```



4. LAPORAN RESMI

Proyek - 3 merupakan kerja individu. Gunakan metode Regresi Logistik Biner, dan bandingkan dengan model regresi logistik lainnya atau metode non statistik sebagai komparasi dan seleksi model terbaik untuk memecahkan masalah klasifikasi. Sumber dataset dapat diambil dari dataset riil (BPS, Open Data Jakarta/Jabar/Surabaya, data kesehatan, dll) atau melakukan survey sendiri.

Luaran dari proyek ini adalah:

1. Laporan Proyek – 3 berbentuk makalah dengan format IEEE dengan konten sesuai dengan poin 3 (studi kasus yang dibahas pada modul ini)
2. File Lampiran dan Source Code