

BAB II

TINJAUAN PUSTAKA

2.1 Logistic Regression

Menurut [11] Logistic Regression adalah suatu teknik klasifikasi yang sering diterapkan. Penerapan Logistic Regression saat menggunakan biner, variabel dependen merupakan variabel dikotomis. Adapun Logistic Regression yang di pergunakan pada multinomial disaat variabel dependen yaitu variabel dengan katagori lebih dari 2 (dua) kategori. Penelitian lain [12] Logistic Regression adalah metode statistic yang menetapkan hubungan matematis antara variabel dependen dan independen serta modelnya.

secara umum model Logistic Regression dapat dirumuskan pada persamaan 2.1

$$\pi(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k}} \quad (2.1)$$

Keterangan :

$\pi(x)$ yaitu probabilitas dari nilai $0 \leq \pi(x) \leq 1$, bahwa Logistic Regression dapat memberikan suatu probabilitas yang bearti, dengan mengubah $\pi(x)$ maka transformasi dengan logit (x) pada persamaan diatas, dirumuskan dengan persamaan 2.2

$$g(x) = \ln \left(\frac{\pi(x)}{1 - \pi(x)} \right) \quad (2.2)$$

bentuk logit yang diperoleh adalah :

$$g(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k \quad (\text{Menurut [13]}) \quad (2.3)$$

Untuk mendapatkan penilaian dari parameter Logistic Regression bisa dilakukan dengan menggunakan cara *Maximum Likelihood Estimation* (MLE) sebagai berikut : penilaian parameter dalam model logit menggunakan *Maximum Likelihood* dengan langkah-langkah sebagai berikut.

1. Fungsi likelihood dari Y

Fungsi likelihood dari Y dirumuskan pada persamaan 2.3

$$L(\beta) = \prod_{i=1}^n [p]^{y_i} [q]^{1-y_i} \quad (2.3)$$

2. (FIL) Fungsi ln-likelihood

Fungsi ln-likelihood dirumuskan pada persamaan 2.4

$$\ln L(\beta) = \ln \left\{ \prod_{i=1}^n [p]^{y_i} [q]^{1-y_i} \right\} \quad (2.4)$$

$$\begin{aligned} \ln L(\beta) &= \ln \left\{ \prod_{i=1}^n [p]^{y_i} [1-p]^{1-y_i} \right\} \\ \ln L(\beta) &= \ln \left\{ \prod_{i=1}^n \left[\frac{\exp(g(x))}{1 + \exp(g(x))} \right]^{y_i} \left[1 - \left(\frac{\exp(g(x))}{1 + \exp(g(x))} \right) \right]^{1-y_i} \right\} \\ \ln L(\beta) &= \sum_{s=0}^q \left\{ \left[\sum_{i=1}^n y_i x_q \right] \beta_q - \sum_{i=1}^n \ln \left[1 + \exp \left(\sum_{s=0}^q \beta_q x_q \right) \right] \right\} \end{aligned}$$

3. (FIL) Fungsi ln-likelihood diturunkan terhadap β

Fungsi ln-likelihood diturunkan terhadap β dirumuskan dengan persamaan 2.5

$$\frac{\partial \ln L(\beta)}{\partial \beta} = \sum_{i=1}^n y_i x_q - \sum_{i=1}^n x_q \left(\frac{\exp(g(x))}{1 + \exp(g(x))} \right) \quad (2.5)$$

$$\frac{\partial^2 \ln L(\beta)}{\partial \beta \partial \beta^T} = \sum_{i=1}^n x_q x_{q^T} p(1-p)$$

4. Jika bentuk close form yang tidak menghasilkan , maka penilaian kriteria β , prosedur iterasi diperoleh melalui cara metode Newton Raphson (NR). Pendekatan metode NR diperoleh dari deret Taylor pada persamaan 2.6

$$\begin{aligned} \frac{\partial \ln L(\beta)}{\partial \beta} + (\beta^{(1)} - \beta) \frac{\partial^2 \ln L(\beta)}{\partial \beta \partial \beta^T} + \frac{1}{2!} (\beta^{(1)} - \beta) (\beta^{(1)} - \beta)^T \frac{\partial^3 \ln L(\beta)}{\partial \beta^T \partial \beta \partial \beta^T} + \dots = 0 \end{aligned} \quad (2.6)$$

Vektor β adalah nilai awal yang sudah ditetapkan. Apabila $|(1) - \beta|$ diperkirakan kecil sekali, maka suku ketiga dan seterusnya bisa diabaikan. Sehingga perluasan atau pengembangan dari deret Taylor dapat ditulis sebagai berikut.

$$\frac{\partial \ln L(\beta)}{\partial \beta} + (\beta^{(1)} - \beta) \frac{\partial^2 \ln L(\beta)}{\partial \beta \partial \beta^T} = 0$$

$$\beta^{(1)} = \beta - \left(\frac{\partial^2 \ln L(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial \ln L(\beta)}{\partial \beta}$$

Secara umum iterasi ke-t metode Newton Raphson adalah sebagai berikut.

$$\beta^{(t)} = \beta^{(t-1)} - \left(\frac{\partial^2 \ln L(\beta)}{\partial \beta^{(t-1)} \partial \beta^{T(t-1)}} \right)^{-1} \frac{\partial \ln L(\beta)}{\partial \beta^{t-1}}$$

Pada saat iterasi melakukan proses, maka akan berhenti di saat $\|\mathbf{t} - \beta(\mathbf{t}-1)\| \leq \epsilon$.

Proses penerapan dilakukan dengan menggunakan pendekatan klasifikasi, dengan membagi menjadi 2 data bagian. Pada data pertama akan diterapkan menjadi data latih, yang digunakan sebagai pembentuk model klasifikasi Logistic Regression. Pada data kedua berikutnya, diterapkan menjadi set validasi terhadap cross validasi sebagai pembentukan fungsi klasifikasi Logistic Regression. Diharapkan dengan dilakukannya pendekatan klasifikasi mampu untuk meminimalis kesalahan-kesalahan dari pengelompokkan, atau meminimalkan rata-rata efek buruk dari kesalahan pengelompokkan, hal ini perlu evaluasi.

2.2 Naïve Bayes

Menurut [14] *Naïve Bayes* adalah metode pengklasifikasi probabilitas yang menghitung sekumpulan probabilitas dengan cara sederhana dalam menjumlahkan frekuensi dan kombinasi jumlah (nilai) dari data set yang diberikan. Algoritma menggunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas. Peneliti lain mengatakan *Naïve Bayes* merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya [15].

Naïve Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai output. Dengan kata lain, diberikan nilai output, probabilitas mengamati secara bersama adalah produk dari probabilitas

individu [16]. Keuntungan penggunaan Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifian. *Naïve Bayes* sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata kompleks dari pada yang diharapkan [17].

2.2.1 Teorema Bayes

Persamaan dari teorema Bayes dapat dirumaskan dengan persamaan 2.7

$$P(H/X) = \frac{P(H/X) \cdot P(H)}{P(X)} \quad (2.7)$$

dimana :

X : Data dengan *class* yang belum diketahui

H : Hipotesis data merupakan suatu *class* spesifik

$P(H|X)$: Probabilitas hipotesis H berdasar kondisi X (posterior probabilitas)

$P(H)$: Probabilitas hipotesis H (prior probabilitas)

$P(X|H)$: Probabilitas hipotesis X berdasar kondisi pada hipotesis H

$P(X)$: Probabilitas hipotesis X

Untuk menjelaskan metode *Naïve Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode *Naïve Bayes* di atas disesuaikan pada persamaan 2.8 sebagai berikut :

$$P(C / F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n / C)}{P(F_1 \dots F_n)} \quad (2.8)$$

variabel C berfungsi menerangkan bentuk dari kelompok (kelas), dimana variabel $F_1 \dots F_n$, menerangkan sebagai bentuk karakteristik yang diperlukan sebagai petunjuk untuk melakukan pengelompokkan. Uraian lebih lanjut rumus *Bayes* tersebut dilakukan dengan menjabarkan $(C|F_1, \dots, F_n)$ menggunakan aturan perkalian sesuai dengan rumus pada persamaan 2.9 sebagai berikut :

$$\begin{aligned} P(C / F_1, \dots, F_n) &= P(C)P(F_1, \dots, F_n / C) \\ &= P(C)P(F_1 / C)P(F_2, \dots, F_n / C, F_1) \\ &= P(C)P(F_1 / C)P(F_2 / C, F_1)P(F_3, \dots, F_n / C, F_1, F_2) \end{aligned}$$

$$\begin{aligned}
&= P(C)P(F_1/C)P(F_2/C, F_1)P(F_3/C, F_1, F_2)P(F_4, \dots, F_n/C, F_1, F_2, F_3) \\
&= P(C)P(F_1/C)P(F_2/C, F_1)P(F_3/C, F_1, F_2) \dots P(F_n, \dots, F_{n-1}/C, F_1, F_2, \dots, F_{n-1}) \quad (2.9)
\end{aligned}$$

Hasil penjabaran tersebut dapat dilihat bahwa semakin banyak dan semakin kompleksnya faktor-faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Disinilah digunakan asumsi independensi yang sangat tinggi (*naïve*), bahwa masing-masing petunjuk (F_1, F_2, \dots, F_n) saling bebas (*independen*) satu sama lain. Dengan asumsi tersebut, maka berlaku suatu kesamaan seperti pada persamaan 2.10

$$P\left(\frac{F_i}{F_j}\right) = \frac{P(F_i \cap F_j)}{P(F_j)} = \frac{P(F_i)P(F_j)}{P(F_j)} = P(F_i) \text{ untuk } i = j, \quad (2.10)$$

Berdasarkan kesamaan diatas, diperoleh rumus pada persamaan 2.11

$$P(F_i | C, F_j) = P(F_i | C) \quad (2.11)$$

persamaan di atas adalah model dari teorema *Naïve Bayes* yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinu digunakan rumus *Densitas Gauss* pada persamaan 2.12 sebagai berikut :

$$P(X_i = x_i | Y = y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}} \quad (2.12)$$

dimana :

P : Peluang

X_i : karakter ke $- i$

x_i : jumlah (nilai) karakter ke $- i$

Y : Kelompok (kelas) yang dicari

y : Sub kelas Y yang dicari

μ_y : *mean*, menerangkan rata-rata dari seluruh karakter

σ_y : Standar deviasi, menerangkan varian dari seluruh karakter

2.2.2 Naïve Bayes Untuk Klasifikasi

Menurut [18] Klasifikasi adalah suatu teknik yang melakukan pencarian sekumpulan model atau fungsi yang menggambarkan dan membedakan kelas data dengan tujuan agar model tersebut dapat digunakan untuk memprediksi kelas dari

suatu objek yang belum diketahui kelasnya. Selain itu, klasifikasi Naïve Bayes terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam basis data dengan jumlah yang besar.

Salah satu theorem Bayes yang digunakan pada klasifikasi adalah *Naïve Bayes*. Klasifikasi *Naïve Bayes* dinyatakan dengan persamaan 2.13 sebagai berikut:

$$v_{MAP} = \arg \max_{v_j \in V} P(a_1, a_2, \dots, a_n | v_j) P(v_j) \quad (2.13)$$

Berdasarkan asumsi pada penyederhanaan *Naïve Bayes*, bahwa jumlah (nilai) aribut berdasarkan conditional saling bebas jika diberikan nilai output. Atau dapat diberikan $P(a_1, a_2, \dots, a_n | v_j) = \prod_i P(a_i | v_j)$ kemudian dimasukkan pada persamaan 2.13, maka akan didapat pendekatan yang dipakai dalam *Naïve Bayes* klasifikasi yang dinyatakan dengan rumus pada persamaan 2.14

$$v_{NB} = \arg \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (2.14)$$

dimana v_{NB} yaitu nilai output hasil klasifikasi Naïve Bayes.

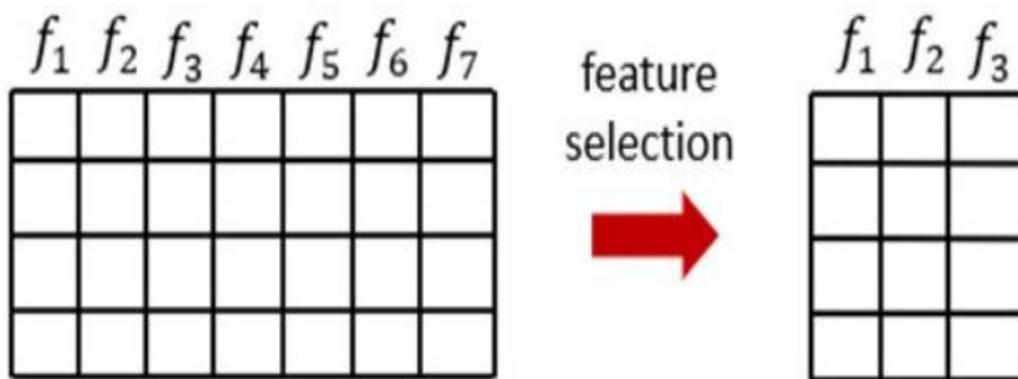
2.3 Metode Feature Selection

Feature selection pada umumnya dapat menghasilkan kinerja pembelajaran yang lebih baik, yaitu, akurasi pembelajaran yang lebih tinggi, biaya komputasi yang lebih rendah, dan kemampuan interpretasi model yang lebih baik[19].

Feature selection adalah, sebagai teknik pengurangan dimensi, bertujuan untuk memilih subset kecil dari fitur yang relevan dari aslinya fitur dengan menghapus fitur yang tidak relevan, berlebihan, atau berisik[20]. Menurut [21] Definisi feature selection adalah mengamati sekumpulan fitur kemudian dipilih beberapa fitur yang mampu memberikan hasil yang terbaik untuk klasifikasi. Metode seleksi fitur digunakan di banyak area aplikasi sebagai alat untuk menghapus yang tidak relevan atau fitur yang berlebihan . Menurut [22] Algoritma Seleksi Fitur digunakan dalam pra-pemrosesan langkah data. Ini mendukung untuk memilih subset yang sesuai dari fitur untuk membangun model untuk data mining. Beberapa masalah pemodelan memiliki sejumlah besar variabel yang dapat memperlambat

pengembangan dan pelatihan model dan memerlukan sejumlah besar memori sistem. Selain itu, kinerja beberapa model dapat menurun ketika memasukkan variabel input yang tidak relevan dengan variabel target.

Metode seleksi fitur berfungsi untuk mengurangi jumlah variabel input menjadi variabel yang diyakini paling berguna bagi model untuk memprediksi variabel target [23]. Pada gambar berikut ini dapat dilihat ilustrasi dari proses atau kinerja daripada seleksi fitur



Gambar 2.1: Ilustrasi pemilihan fitur dan pengurangan ukuran data pada data tabular [23].

Pada gambar diatas yang berkaitan dengan apakah fitur dipilih berdasarkan variabel target atau tidak. Teknik pemilihan fitur tanpa pengawasan mengabaikan variabel target, seperti metode yang menghilangkan variabel yang berlebihan menggunakan korelasi. Teknik pemilihan fitur yang dikontrol menggunakan variabel target, seperti metode yang menghilangkan variabel yang tidak relevan.

2.4 Metode forward Selection

Operator Forward Selection adalah operator bersarang yaitu memiliki subproses. Subproses dari operator Forward Selection harus selalu mengembalikan performance vector [24]. Menurut [25]. Forward selection digunakan untuk memilih atribut yang memenuhi kriteria, jadi hanya atribut terpilih yang akan masuk ke proses klasifikasi. Diharapkan pemilihan atribut menggunakan Forward Selection dapat mengatasi masalah ketidakseimbangan kelas dan meningkatkan prediksi ketepatan.

Operator Forward Selection (OFS) dimulai dengan pemilihan atribut yang kosong dan, di setiap putaran, ia menambahkan setiap atribut yang tidak digunakan dari ExampleSet yang diberikan. Pada metode ini, satu variabel dipilih sebagai target, dan variabel independen yang mempengaruhi variabel target ini yang akan dicari. Metode pemilihan forward Selection cukup sederhana dan cepat, hal ini dapat dilakukan sebagai berikut [26]:

- Lakukan regresi linier satu dimensi dengan setiap kandidat regresi (dari n)
- Temukan yang paling signifikan (dengan membandingkan kriteria kualitas R^2). Kami sebut saja X_A , dan parameter terkait disebut B_A
- Hitung variabel independen baru (y_A)

Rumus untuk menghitung variabel independen baru (y_A) terdapat pada persamaan 2.15

$$y_A = y - x_A \theta_A \quad (2.15)$$

yaitu bagian dari y yang dapat dijelaskan secara linier oleh X_A dikurangi dari y

- Lakukan regresi linier satu dimensi dengan setiap regresi yang tersisa kandidat (dari $n - 1$)

Find the most significant one: We call it X_B , with parameter B_B

- Hitung variabel independen baru (y_B)

Rumus untuk menghitung variabel independen baru (y_B) terdapat pada persamaan 2.16

$$y_B = y_A - x_B \theta_B \quad (2.16)$$

- Lanjutkan dengan cara ini, sampai sejumlah $n-1$ regressor dipilih

Prosedur ini cepat, karena pada langkah i hanya $n - i - 1$ regresi satu dimensi harus dihitung. Tetapi prosedur ini tidak memperhitungkan setiap interaksi antara regressor. Oleh karena itu, varian dengan ortogonalisasi telah dibangun

2.5 Cross-validation

Cross-validation (CV) adalah salah satu metode resampling data yang paling banyak digunakan untuk memperkirakan prediksi yang sebenarnya, kesalahan

model dan untuk menyesuaikan parameter model [27]. Menurut [28] Cross-validation menggunakan subset data acak yang dikenal sebagai k-folds, cross-validation adalah cara yang efektif untuk menguji tingkat keberhasilan model yang digunakan untuk klasifikasi.

Pada penulisan ini, penulis menggunakan k-fold 10, dimana data dibagi menjadi dua subset, yaitu data latih untuk proses pembelajaran dan data uji untuk validasi atau evaluasi, yang digunakan untuk mengevaluasi kinerja model atau metode atau algoritma, untuk memberikan nilai k disarankan tidak terlalu besar dan tidak terlalu kecil. Menurut [29] Nilai k yang terlalu besar akan menghasilkan model yang tidak bias, tetapi dapat membuat variansi menjadi besar sehingga dapat memicu terjadinya overfit. Nilai k yang terlalu kecil akan menghasilkan model yang serupa dengan metode Cross-validation biasa yang hanya membagi data menjadi train – test saja (dapat memicu terjadinya bias). Nilai k yang biasa digunakan adalah $k = 5$ atau $k = 10$. Sedangkan [30] Salah satu metode Cross-validation yang akan digunakan dalam hal analisis, sebanyak percobaan pada klasifikasi menggunakan nilai $k = 10$ menjadi pilihan terbaik untuk proses validasi yang sangat tepat.

2.6 Hasil Penelitian yang Relevan

Berikut adalah beberapa hasil penelitian yang menjadi referensi dan memberikan banyak masukan kepada penulis :

Tabel 2.1 Relevansi

No	Judul	Tahun	Metode	Dataset	Hasil
1	“COVID-19 Patient Health Prediction Using Boosted Random Forest Algorithm”[31].	2020	Random Forest (RF) model boosted by the AdaBoost	The dataset used accessed from Kaggle as Novel Corona Virus 2019 Dataset	Random Forest (Rf) algorithm booted by the Adaboost algorithm, with a f1 score of 0.86
2	“Impact of SARS-CoV-2 in Online Education, Predicting and	2021	Clustering The K-means Classification Random Forest	The dataset has been compiled from 647 understudies’rea	Rf accuracy = 83.3 KNN accuracy = 55.18

	Contrasting Mental Stress of Young Students: A Machine Learning Approach” [1].		KNN, Logistic regression, Naive Bayes, ID3 ,SVM	ctions to online course suggestions. .	SVM accuracy = 87.75 Lr accuracy = 79.13 Nb accuracy = 71 Dt accuracy = 54.14
3	“Covid-19 and its impact on school closures: a predictive analysis using machine learning algorithms” [32].	2021	Decision Tree (DT) and ANN	https://en.unesco.org/covid19/educationresponse In this study, the UNESCO the dataset contains 92400	ML models where ANN displayed maximum accuracy of 80.37%. After rigorous comparative analysis, Decision Tree (DT) portrayed the highest accuracy of 90.75%.
4	“K- means Clustering of Online Learning Profiles of Higher Education Teachers and Students Amid Covid19 Pandemic” [33].	2020	K-means clustering algorithm was used to group the respondents according to the similarities of their traits	These 420 respondents target profile.	It is shown that among the 420 respondents, 217 (51.67%) belong to the Movers, 148 (35.24%) belong to the Trailers, and 55 (13.09%) belong to the Catchers.
5	“Classifying the Parental Involvement on School From Home	2020	Demographic factors after conducting the	data that has been collected is including the child age and	accuracy of the C4.5 algorithm in classifying parental involvement is

	during Covid-19 using C4.5 Algorithm “ [30].		data reprocessing are used as input in the C4.5 algorithm	gender, parents age, parent education level, parent occupation, parent earning, number of siblings, and number of family members at home	86.67%. This result is good enough to be used in the classification Process .
6	“Komparasi Algoritma Naïve Bayes dan Algoritma K-Nearst Neighbor terhadap Evaluasi Pembelajaran Daring “[10].	2021	model Algoritma Naïve Bayes dan Algoritma K-Nearst Neighbor menggunakan tools Rapidminer Versi 9.3. dengan <i>operator retrieve, Cross-validation,</i> Algoritma Naïve Bayes atau K-Nearst	data yang digunakan data primer artinya data di ambil dari hasil sebaran kuisioner pada mahasiswa semester 3 (tiga) semester 5 (lima) dan semester 7 (tujuh) sebanyak 352 responden	Hasil ketepatan Algoritma Naïve Bayes dengan angka 91, 45%. Hasil ketepatan Algoritma K-Nearst Neighbor dengan angka 97, 72%.
7	“Comparative Analysis of Stress among Undergraduate	2021	In this paper till now we have used two algorithms that	We received about 109 responses but after filtering we	The logistic regression and random forest

	Students Using Logistic Regression and Random Forest Techniques” [8].		are random forest and logistic regression	were left with only 101.	algorithm, where we got our f1 score = 0.9411 and Accuracy score (Random Forest) = .0.8571
8	“Mental Stress Detection in University Students using Machine Learning Algorithms” [9].	2021	Classification Algorithm Random Forest, Naïve Bayes, SVM and KNN	The dataset was taken from Jaypee Institute of Information Technology and it consisted of 206 student’s data	We found that support vector algorithms giving an accuracy of 85.71%, specificity 100%, and sensitivity of 75%, Random forest is performing next to support vector machine giving an accuracy of 83.33%, specificity of 66.66%, and sensitivity of 100%. Thus we can say that SVM is performing well out of these four algorithms in this scenario
9	“Impact on Human	2021	Naive Bayes classifiers,	For data collection, we	NB accuracy = 88 %

	Mental Behavior after Pass through a Long Time Home Quarantine Using Machine Learning”. [34]		Support Vector Machine, and logistic regression.	have created a Google Form and we post it on social media	SVM accuracy = 71 % Lr accuracy = 76 %
10	“Prediction and Analysis of Data Mining Models for Students Underlying Issues during Novel Coronavirus (COVID-19)” [35]	2020	random forest , decision tree , support vector machine , logistic regression, K-nearest neighbor , naive Bayes	Dataset instances of the student’s problem were considered.	DT accuracy = 94.85 SVM accuracy = 92.85 Lr accuracy = 87.52 NB accuracy = 90.49 KNN accuracy = 89.60 Rf accuracy = 88.06
11	Comparison of the Performance of Machine Learning-based Algorithms for Predicting Depression and Anxiety among University Students in Bangladesh: A Result of the First Wave of the COVID-19 Pandemic[36]	2021	This study applied six well-known ML algorithms, namely logistic regression, random forest (RF), support vector machine (SVM), linear discriminate analysis, K-NN , Naïve Bayes,.	A structured questionnaire-based online survey was conducted on 2121 university students (private and public) living in Bangladesh	RF (91.3% accuracy), SVM (91.49% accuracy). LR (78.07% accuracy), KNN (91.3% accuracy), LDA (79.58% accuracy), NB (73.53% accuracy),