JINTEKS (Jurnal Informatika Teknologi dan Sains) ISSN 2686-3359 (Online)
Vol. 5 No. 1, Februari 2023, hlm. 18 – 24

PERBANDINGAN ALGORITMA *NAÏVE BAYES* DAN SVM UNTUK ANALISIS PENYALAHGUNAAN KEJAHATAN *CARDING*

Putri Ramayanti[1], Tata Sutabri[2]
[1,2]*Program* Studi Magister Teknik Informatika, Universitas Bina Darma Palembang
email: putriramabidar@gmail.com

Abstrak: *Carding* merupakan salah satu wujud aksi kejahatan siber(*cybercrime*) yang berkaitan dengan dunia perbankan ialah kartu kredit. Kejahatan tersebut merupakan metode pencurian no kartu kredit dari web sah ataupun *spammer* serta memakainya demi keuntungan pribadi. Permasalahan penipuan ini susah diidentifikasi sehingga yang sering dirasakan oleh korban *carding* merupakan kemunculan rasa ragu yang terus menerus dikala hendak memakai sistem belanja kartu kredit. Berdasarkan permasalahan ini, maka peneliti ingin dapat menganalisa kejahatan *carding* tersebut menggunakan dua algoritma *machine learning* yaitu *Naïve Bayes* dan *Support vector machine* sehingga dapat membantu mendeteteksi kejahatan *carding*. Berdasarkan hasil penelitian dapat disimpulkan bahwasanya penyalahgunaan kejahatan *carding* dapat di prediksi secara akurat baik itu menggunakan Algoritma *Naïve Bayes* dan *Support vector machine*. Akan tetapi Algoritma Support Machine lebih akurat daripada *Naïve Bayes*. Algoritma Support Machine mendapatkan skor 99% lebih tinggi 1% daripada *Naïve Bayes*. Hal ini dikarenakan Pengklasifikasi *Support vector machine* menawarkan akurasi tinggi dan bekerja dengan baik.

Kata Kunci : *Carding, Machine Learning, Naïve Bayes,* dan *Support vector machine.*

Abstract: *Carding is a type of cybercrime related to banking, or credit cards. This crime is a method of stealing credit card numbers from legitimate websites and spammers and using them for personal gain. Due to the difficulty in identifying this fraud issue, card victims often experience lingering suspicions when attempting to use credit card buying schemes. Based on this problem, researchers hope to be able to analyze card crime using two machine learning algorithms that help detect card crime: Naive Bayes and Support vector machine. Based on the results of our research, we can conclude that both the Naive Bayes algorithm and Support vector machines can be used to accurately predict carding crime exploits. However, support machine algorithms are more accurate than Naive Bayes. The Support Machine Algorithm produces 1's values that are 99% higher than Naive Bayes. This is because support vector machine classifiers provide high accuracy and work well.*

Keywords **:** *Carding, Machine Learning, Naïve Bayes, and Support vector machine.*

PENDAHULUAN

Akhir-akhir ini Kenaikan kegiatan transaksi di platform digital sepanjang masa pandemi ikut mengerakan kegiatan transaksi memakai kartu kredit. Berlanjutnya tren kenaikan transaksi digital ini bakal mengimbangi tekanan di lini travel yang lebih dahulu jadi pasar utama layanan kartu kredit.

Tidak hanya itu, kenaikan transaksi digital juga bisa mengimbangi kemampuan tekanan pada transaksi kartu kredit yang bisa jadi terjalin akibat pembatasan mobilitas sejak *pandemic covid*-19.

Bersumber pada informasi Bank Indonesia, volume belanja kartu kredit bulanan meningkat secara tahunan sejak 2021. Semakin meningkatnya volume aktivitas kartu kredit maka mengakibatkan munculnya tindak kejahatan yang sering disebut *cyber-crime*. Dan salah diantaranya kejahatan *carding.* pembobolan kartu kredit lewat *carding* merupakan salah satu contoh yang kerap muncul. Tidak tanggung jumlah korbannya sangatlah banyak.

Carding merupakan salah satu wujud aksi kejahatan siber(*cybercrime*) yang berkaitan dengan dunia perbankan ialah kartu kredit. Kejahatan tersebut merupakan metode pencurian no kartu kredit dari web sah ataupun *spammer* serta memakainya buat pembelian gift card prabayar.

Kartu gift itu setelah itu dijual kembali dengan tujuan buat memperoleh keuntungan dalam nominal uang yang sangat besar. Apabila dipaparkan dalam dunia nyata hingga permasalahan kejahatan *carding* merupakan permasalahan pencurian. Bagi salah satu unit FBI ialah IFFC(*Internet Fraud Complaint Centre*), *carding* merupakan pemakaian yang tidak legal dari kartu kredit ataupun kartu debit *fraudulently*.

Dilansir dari banyak sumber, aktivitas ini bertujuan buat mendapatkan keuntungan berupa uang ataupun properti di mana kartu kredit ataupun no kartu debit bisa dicuri ataupun dibeli dari web website palsu atau bia disebut juga *scheme*.

Bisa dikatakan kalau penyalahgunaan kartu ataupun *carding* merupakan modus sangat simpel sebab owner kartu kurang waspada, sehingga pelaku kejahatan bisa memanfaatkan hal tersebut untuk melakukan tindak kriminal *carding*.

Di samping itu, banyak pula warga Indonesia yang belum mengerti menimpa apa itu *carding* walaupun di Indonesia sendiri permasalahan satu ini telah lumayan kerap terjalin.

Permasalahan penipuan ini susah diidentifikasi sehingga yang sering dirasakan oleh korban *carding* merupakan kemunculan rasa ragu yang terus menerus dikala hendak memakai sistem belanja kartu kredit.

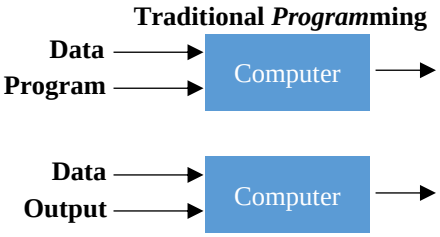
 ISSN 2686-3359 (Online)

Berdasarkan permasalahan ini, maka peneliti akan mencoba untuk mendeteksi penyalah gunaan kejahatan karding dengan menggunakan machine learning. *Machine learning* memiliki banyak sekali algoritma. Maka peneliti akan mencoba menemukan algoritma mana yang paling effective untuk permasalaan tersebut. Algoritma yang akan dibandingkan adalah algoritma *naïve bayes* dan *support vector machine*.

TINJAUAN PUSTAKA

Machine Learning ialah salah satu cabang dari ilmu kecerdasan buatan, spesialnya yang menekuni tentang gimana komuputer sanggup belajar dari informasi buat tingkatkan kecerdasannya.

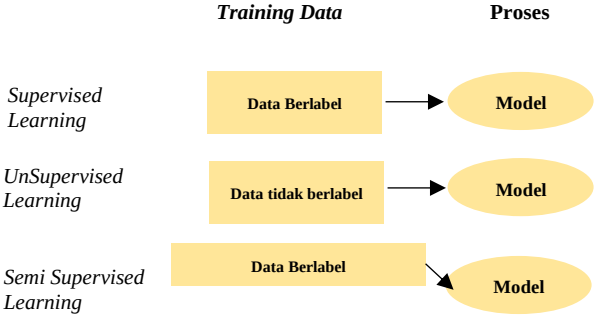
Machine learning mempunyai fokus pada pengembangan suatu sistem yang sanggup belajar sendiri buat memutuskan suatu, tanpa wajib kesekian kali de*program* oleh manusia. Dengan begitu mesin tidak cuma dapat menciptakan ketentuan buat perlikau maksimal dalam pengambilan keputusan, tetapi pula dapat menyesuaikan diri dengan pergantian yang terjalin. Dalam pendidikan mesin, mesin mengenalisis kumpulan informasi yang besar buat menciptakan pola.

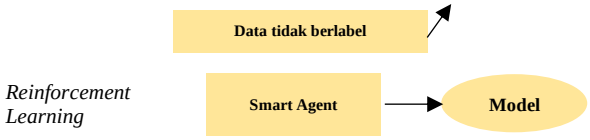


Gambar 1. Perbedaan Pemrograman Tradisional dan *Machine Learning*

Penulis melalui Algoritma pendidikan mesin butuh melatih komputer sedemikian rupa sehingga dapat menguasai *model* objek yang dikenali manusia. Pada pemrograman traditional, informasi serta *program* dijalankan pada pc buat menciptakan *output*, sebaliknya pada pemrograman memakai *machine learning*, informasi serta *output* dijalankan pada komputer buat membuat *program*, serta setelah itu *program* tersebut digunakan dalam pemrograman tradisional.

Istilah inggris dimiringkandikategorikan jadi 4 jenis. Keempat jenis tersebut merupakan *supervised learning* ataupun algoritma terbimbing, un*supervised learning* ataupun algoritma tidak terbimbing, semi *supervised learning* serta reinforcement learning.





Gambar 2. Tipe-tipe *Machine Learning*

Berikut merupakan uraian terperinci untuk jenis *Machine Learning* Supervised. *Supervised Learning* merupakan Metode *Machine Learning* dengan memakai informasi latih buat melakukan pembelajaran. informasi latih tersebut merupakan informasi yang telah berlabel. Tujuancara ini merupakan untuk mengenali label *input* yang baru dengan memakai fitur yang ada agar dapat menghasilkan prediksi ataupun klasifikasi. Sebagian algoritma yang tercantum dalam *supervised learning* merupakan regresi liner bergkita, decision tree, random forest, *naïve bayes classifier*, nearest neighbor, *support vector machine*, serta artificial neural network.

Penulis akan membahas lebih detail *Naïve Bayes classifier* dan *support vector macine*. Dikarenaka di penilitian ini, penulis akan membandingkan kedua algoritma tersebut untuk menganalisa kejahatan *carding*.

Berikut adalah penjelasan detail kedua algoritma tesebut :

1. *Naïve Bayes Classifier*

Algoritma *Naive Bayes* adalah salah satu algoritma pembelajaran mesin yang paling penting untuk membantu masalah klasifikasi. Itu berasal dari teori probabilitas Bayesian dan digunakan untuk klasifikasi teks, di mana data dimensi tinggi dilatih. Beberapa contoh terbaik dari algoritme *Naive Bayes* adalah analisis sentimen, klasifikasi artikel baru, dan pemfilteran *spam*. Algoritma klasifikasi digunakan untuk mengklasifikasikan pengamatan baru ke dalam kelas yang telah ditentukan sebelumnya untuk data yang tidak diinisialisasi. Algoritma *Naive Bayes* dikenal karena kesederhanaan dan efisiensinya. Algoritma ini mempercepat pembuatan *model* dan membuat prediksi. Saat membuat *model* ML, lebih baik menerapkan teorema Bayes. Menerapkan algoritme *Naive Bayes* memerlukan keterlibatan developer ML yang berpengalaman

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

Ini adalah algoritma yang memeriksa probabilitas setiap objek, propertinya, dan termasuk grup mana. Ini juga dikenal sebagai pengklasifikasi probabilistik. Algoritma *Naive Bayes* tunduk pada pembelajaran yang diawasi dan terutama digunakan untuk memecahkan masalah klasifikasi.

Probabilitas adalah dasar dari algoritma *Naive Bayes*. Algoritma ini didasarkan pada hasil

 ISSN 2686-3359 (Online)

probabilistik yang dapat diberikannya untuk masalah yang belum terpecahkan dengan prediksi. Berikut adalah detail lebih lanjut tentang probabilitas, teori Bayesian, dan probabilitas bersyarat:

1.1 *Probabilitas*

Probabilitas membantu memprediksi terjadinya suatu peristiwa dari semua kemungkinan hasil. Persamaan matematika untuk probabilitas adalah sebagai berikut:

$$Probabilitas\ peristiwa\ = \frac{jumlah\ peristiwa\ yang\ menguntungkan}{jumlah\ total\ hasil}$$

0 < = probabilitas kejadian < = 1. Hasil positif berarti kejadian karena probabilitas. Probabilitas selalu antara 0 dan 1, di mana 0 berarti probabilitas suatu peristiwa tidak mungkin terjadi dan 1 berarti tingkat keberhasilan suatu peristiwa mungkin terjadi. Untuk pemahaman yang lebih baik, juga dapat mempertimbangkan kasus di mana memprediksi buah berdasarkan warna dan teksturnya. Berikut adalah beberapa kemungkinan asumsi yang dapat dibuat. Kita dapat memilih buah yang tepat yang Kita pikirkan, atau bingung dengan buah yang serupa dan membuat kesalahan. Bagaimanapun, kemungkinan memilih buah yang benar adalah 50%.

1.2 Bayes Theory

Teori Bayesian mencoba untuk sampai pada hipotesis (H) dari serangkaian bukti (E) yang diberikan. Itu bermuara pada dua hal yaity probabilitas hipotesis sebelum bukti P(H) dan probabilitas setelah bukti P(H|E). Teori Bayesian dijelaskan dengan persamaan berikut:

$$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$$

- P(H|E) menunjukkan bagaimana peristiwa H terjadi ketika peristiwa E terjadi.
- P(E|H) menyatakan berapa kali kejadian E terjadi jika kejadian H terjadi terlebih dahulu.
- P(H) menyatakan probabilitas kejadian X akan terjadi dengan sendirinya.
- P(E) menyatakan probabilitas kejadian Y akan terjadi dengan sendirinya.

Aturan Bayes adalah metode untuk menentukan P(H|E) dari P(E|H). Singkatnya, ini memberi kita cara untuk menghitung probabilitas hipotesis berdasarkan bukti yang diberikan.

1.3 Conditional Probabilitas

Probabilitas bersyarat adalah bagian dari probabilitas. Ini mengurangi kemungkinan menjadi kecanduan satu peristiwa. Kita dapat menghitung probabilitas bersyarat untuk dua peristiwa atau lebih.Jika Kita mengambil peristiwa X dan Y, probabilitas bersyarat dari peristiwa Y didefinisikan sebagai probabilitas peristiwa yang terjadi ketika peristiwa X telah berlalu. Ditulis sebagai P(Y|X). Rumus matematika untuk ini adalah sebagai berikut:

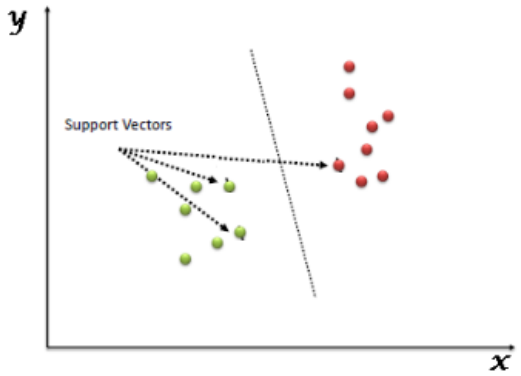
$$P(Y|A) = \frac{P(X\ and\ Y)}{P(X)}$$

Dengan data pelatihan, kita dapat melatih *model* dan menjalankannya. kemudian kita perlu memvalidasi data untuk mengevaluasi *model* dan membuat prediksi baru. Terakhir,kita harus menamai atribut *input* "bukti" dan melabelinya sebagai "*output*" dalam data pelatihan.

Menggunakan probabilitas bersyarat, dilambangkan sebagai P(E|O), kita dapat menghitung probabilitas pembuktian dari keluaran yang diberikan. Tujuan akhir kita adalah untuk menghitung P(O|E) - probabilitas keluaran berdasarkan atribut saat ini. Jika soal memiliki dua hasil, kita dapat menghitung probabilitas setiap hasil dan mengatakan siapa yang menang. Di sisi lain, jika Anda memiliki banyak atribut *input*, diperlukan algoritma *Naive Bayes*.

2. Support vector machine

Support vector machine (SVM) adalah algoritma pembelajaran mesin terawasi yang dapat digunakan untuk tantangan klasifikasi dan regresi. Namun, ini terutama digunakan dalam masalah klasifikasi. Dalam algoritma SVM, kita dapat merepresentasikan setiap titik data sebagai titik dalam ruang n-dimensi (di mana n adalah jumlah fitur tertentu yang kita miliki), di mana nilai setiap fitur adalah nilai koordinat tertentu. Kemudian kita melakukan klasifikasi dengan mencari *hyperplane* yang memisahkan kedua class dengan sangat baik (seperti gambar dibawah ini).



Gambar 3. Gambar klasifikasi SVM

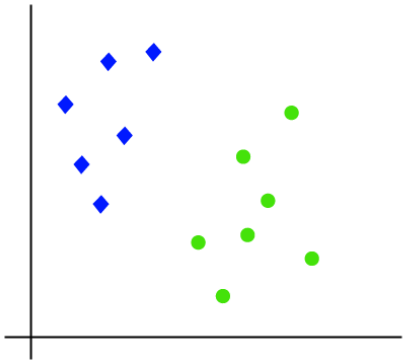
Vektor pendukung hanyalah koordinat dari satu pengamatan. Klasifikasi SVM adalah batas terbaik yang memisahkan dua kelas (*hyperplane*/line).

Berikut adalah cara kerja SVM

1. Linear SVM

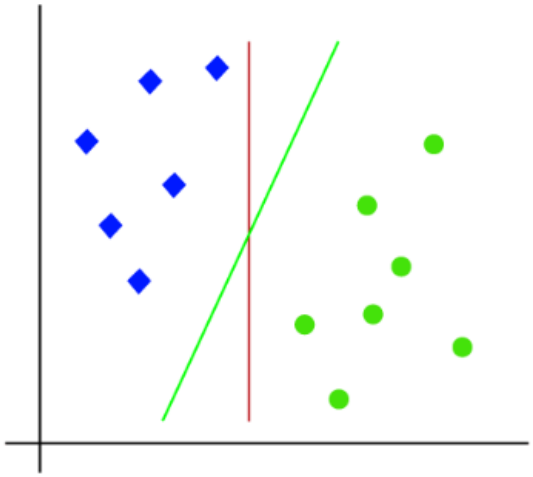
Pengoperasian algoritma SVM dapat dipahami dengan sebuah contoh. Misalkan kita memiliki kumpulan data dengan dua label (hijau dan biru) dan kumpulan data tersebut memiliki dua fitur x1 dan x2. Kita menginginkan pengklasifikasi yang dapat mengklasifikasikan sepasang koordinat (x1,x2) sebagai hijau atau biru. Kita bisa melihat dari gambar di bawah ini:

 ISSN 2686-3359 (Online)



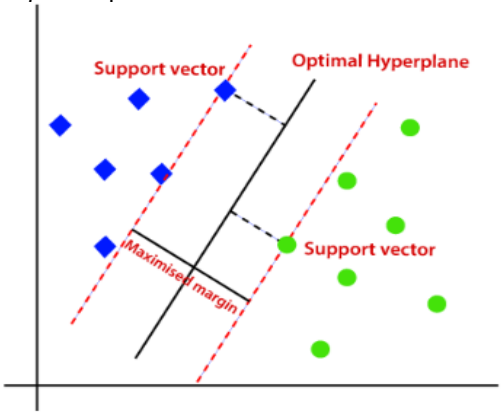
Gambar 4. ruang 2-D

Karena ini adalah ruang 2-D, kita dapat dengan mudah memisahkan kedua kelas dengan garis lurus. Namun, ada beberapa baris yang dapat memisahkan kelas-kelas ini. Perhatikan gambar di bawah ini:



Gambar 5. algoritma SVM

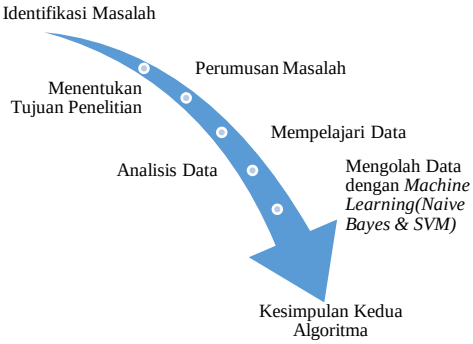
Dengan demikian, algoritma SVM membantu menemukan garis atau batas keputusan terbaik; Batas atau wilayah terbaik ini disebut *hyperplane*. Algoritma SVM mencari titik terdekat dari garis kedua kelas. Titik-titik ini disebut vektor dukungan. Jarak antara vektor dan *hyperplane* disebut margin. Dan tujuan SVM adalah memaksimalkan margin tersebut. *Hyperplane* dengan tepi maksimum disebut *hyperplane* optimal.



Gambar 6. Support Vector

METODE

Penelitian ini menggunakan metode penelitian dengan beberapa tahap proses penelitian seperti yang terlihat pada gambar berikut:



Gambar 7. Desain Penelitian

Berikut ini merupakan uraian dari desain riset yang terdapat pada gambar 11 di atas:1.Identifikasi Permasalahan: Didalam sesi ini riset dimulai dengan melaksanakan penelitian pendahuluan untuk mengenali kasus yang berkaitan dengan topik yang diteliti agar penulis memperoleh apa yang sebetulnya jadi permasalahan untuk dipecahkan. 2. Formulasi Permasalahan: Didalam sesi ini periset merumuskan permasalahan yang sudah didapatkan supaya masalah tersebut bisa dijawab dengan baik melalui riset. 3. Memastikan Tujuan Riset: Pada sesi ini tujuan penelitian ialah mengenali gimana *machine learning* mengetahui tindak kejahatan *carding* memakai 2 algortima yang berbeda.4. Mempelajari Data: Pada sesi ini penulis mencari dan menekuni sumber-sumber pengetahuan berbentuk buku-buku teori, jurnal- jurnal riset, serta sumber pustaka otentik yang lain yang berkaitan dengan penelitian.5. Analisis Data: menganalisa data- data yang berkaitan dengan tindak kejahatan *carding* didapatkan lewat sumber data di internet yakni *Kaggle*. 6. Mengolah Data dengan *Machine Learning* menggunakan dua algoritma yaitu *Naïve Bayes* dan SVM :Data-data yang telah dianalisa setelah itu diolah menggunakan Dua algoritma sebelum merumuskan hasil.7. Kesimpulan: Sesi terakhir dalam penelitian ini ialah merumuskan hasil penelitian yang berisi jawaban terhadap rumusan permasalahan bersumber pada data- data yang terdapat. Periset pula membagikan saran.

Sumber data yang digunakan adalah data kejahatan *carding* yang bersumber dari Kaggle. Dimana sudah terdapat label dimana pelanggan yang yang mengalami kejahatan dan yang bukan mengalami kejahatan *carding*.

HASIL DAN PEMBAHASAN

Untuk melatih, memvalidasi, dan menguji *model*, penulis membuat kumpulan data terdiri dari 1296675 data *carding*. Data *carding* ini dulu

 ISSN 2686-3359 (Online)

dikumpulkan dari dataset yang bersumber dari Kaggle. Berikut table total dataset.

Tabel 1. Dataset *Carding* yang dikumpulkan

Total Sample	1296675
Total training sample	1037340
Total *testing* sample	259335

Berikut adalah langkah-langkah melakukan pemrograman untuk data diatas.

1. Memanggil *library* yang dibutuhkan

Berikut adalah kode *program* untuk memanggil *library* yang dibutuhkan .

```
In [1]:
#memanggil library
import pandas as pd
from sklearn import preprocessing
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.model_selection import train_test_split
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.naive_bayes import GaussianNB
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn import svm
```

Gambar 8. Memanggil *Library*

Library yang digunakan adalah *sklearn* dimana sudah tersedia algoritma *naïve bayes* & *support vector machine*.

2. Memanggil *dataset carding* yang bersumber dari Kaggle

Berikut adalah kode *program* untuk memanggil dataset *carding* dan memasukan ke dalam variable 'df'.

```
In [2]:
#memanggil Dataset karding dari sumber data kaggle
df=pd.read_csv("D:/kuliah/jupyter files/karding.csv")
df = df.drop("trans_date_trans_time", axis='columns')
df = df.drop("unix_time", axis='columns')
df = df.drop("Unnamed: 0", axis='columns')
df=df.dropna()
df.head()
```

Out[2]:

	cc_num	merchant	category	amt	first	last	gender	street	city	state	zip	lat	long	city_pop
0	2703186189652095	fraud_Rippin, Kub and Mann	misc_net	4.97	Jennifer	Banks	F	561 Perry Cove	Moravian Falls	NC	28654	36.0788	-81.1781	3495
1	630423337322	fraud_Heller, Gutmann and Zieme	grocery_pos	107.23	Stephanie	Gill	F	43039 Riley Greens Suite 393	Orient	WA	99160	48.8878	-118.2105	149

Gambar 9. Memanggil *dataset*

Data dipanggil menggunakan *library pandas* dimana *library* tersebut umumnya dipakai untuk pengolahan data dalam bentuk table.

3. Mengubah semua data kedalam bentuk bilangan

Berikut adalah kode *program* untuk mengubah semua data ke dalam bilangan. Agar komputer dapat memproses semua data tersebut.

```
In [4]:
#encode semua kolom data
df_encode=df[['cc_num','merchant','category','amt','first','last','gender','street','city','zip','lat','long','city_pop','job','dob','trans_num','merch_lat','merch_long','is_fraud']]
i=0
for column in df:
    le = preprocessing.LabelEncoder()
    # Converting string labels into numbers.
    nama=df.columns[i]
    test=df.iloc[:, i]
    df_encode[nama]=le.fit_transform(test)
    i+=1
df_encode['is_fraud']=df['is_fraud']
df_encode.head()
```

```
C:\Users\r00262091\AppData\Local\Temp\ipykernel_15248\1675635666.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_encode[nama]=le.fit_transform(test)
```

Out[4]:

	cc_num	merchant	category	amt	first	last	gender	street	city	zip	lat	long	city_pop	job	dob	trans_num	merch_lat	merch_long	is_fraud
0	444	514	8	397	182	18	0	568	526	265	291	693	458	370	779	56438	380185	861455	0
1	42	241	4	10823	339	157	0	435	612	965	964	60	43	428	607	150305	1241513	78062	0

Gambar 10. *Encoding* semua *dataset*

Pada dasarnya komputer hanya mengenal bilangan. Oleh karena data yang digunakan dalam bentuk teks maka harus di ubah dulu dalam bentuk bilangan.

4. Memilah data training dan data *testing*

Selanjutnya pada dataset perlu dilakukan pemisahan data training sebesar 80% dan data *testing* sebesar 20%.

```
#memilah data training & data testing dari data yang telah dipilih
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)
```

Gambar 11. Pemilahan *data training* dan *testing*

Untuk mengukur keberhasilan suatu *model* atau algoritma adalah dengan cara mengukur tingkat akurasi. Oleh karena itu data harus dipisah menjadi dua bagian yaitu *training* dan *testing*.

5. Memanggil *Naïve Bayes Classifier*

Langkah selanjutnya memanggil *naive bayes classifier* dan kemudian melakukan fiting *model* dengan data *training* serta target.

```
In [10]:
#memanggil naive bayes classifier dan melatih data training
gnb = GaussianNB()
gnb.fit(X_train, y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

```
Out[10]: GaussianNB()
```

Gambar 12. Memanggil *naïve bayes*

Program diatas digunakan untuk membuat *model naïve bayes* dengan data *training*.

6. Melakukan Prediksi pada Data *Testing* menggunakan *naïve bayes*

Berikut adalah langkah prediksi menggunakan *naïve bayes*

 ISSN 2686-3359 (Online)

```
In [12]: #memprediksi data testing dengan naive bayes
         y_pred = gnb.predict(X_test)
```

Gambar 13. Prediksi Data *Testing*

Setelah *model* berhasil dibuat setelah itu akan dilakukan *testing* agar mengetahui tingkat akurasinya.

7. Menghitung Akurasi *naïve bayes*

Berikut adalah tingkat akurasi dari *model* prediksi *naïve bayes* yang telah dibuat. Skor akurasi *model* dengan *naïve bayes* adalah 98%

```
In [13]: #melihat skor akurasi dengan naive bayes
         print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

         Accuracy: 0.9802224921433667
```

Gambar 14. Skor Akurasi *Naïve Bayes*

Gambar diatas menunjukan tingkat akurasi *model naïve bayes*. Didapatkan hasil pengukuran 98.02%.

```
In [14]: #mencetak laporan akurasi dengan naive bayes
         print(classification_report(y_test, y_pred))

                       precision    recall  f1-score   support

                    0       1.00      0.98      0.99    257815
                    1       0.16      0.57      0.25      1520

             accuracy                           0.98    259335
            macro avg       0.58      0.78      0.62    259335
         weighted avg       0.99      0.98      0.99    259335
```

Gambar 15. Perhitungan Akurasi *Naïve Bayes*

Gambar diatas menunjukan tingkat akurasi *model naïve bayes* sampai dengan 98%

8. Memanggil Linear SVM Classifier

Langkah selanjutnya memanggil *Linear SVM classifier* dan kemudian melakukan fiting *model* dengan data *training* serta target.

```
In [*]: #memanggil support vector machine classifier dan melatih data training
        clf = svm.SVC(kernel='linear')
        clf.fit(X_train, y_train)
```

Gambar 16. Memanggil *Linear SVM Classifier*

Program diatas digunakan untuk membuat *model Support Vector Machine* dengan data *training*

9. Melakukan Prediksi pada Data *Testing* menggunakan Linear SVM

Berikut adalah langkah prediksi menggunakan *Linear SVM*.

```
:  #memprediksi data testing dengan support vector machine classifier
   y_pred = clf.predict(X_test)
```

Gambar 17. Prediksi Data *Testing*

Gambar diatas adalah *program* untuk memprediksi data.

10. Menghitung Akurasi Linear SVM

Berikut adalah tingkat akurasi dari *model* prediksi Linear SVM yang telah dibuat. Skor akurasi *model* dengan Linear SVM adalah 99%.

```
#melihat skor akurasi dengan support vector machine classifier
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

Accuracy: 0.9937164098275247
```

Gambar 18. Skor Akurasi Linear SVM

Gambar diatas menunjukan tingkat akurasi *model Support Vector Machine*. Didapatkan hasil pengukuran 99.37%.

```
In [19]: #mencetak laporan akurasi dengan support vector machine classifier
         print(classification_report(y_test, y_pred))

                       precision    recall  f1-score   support

                    0       0.99      1.00      1.00   1160249
                    1       0.23      0.04      0.06      6759

             accuracy                           0.99   1167008
            macro avg       0.61      0.52      0.53   1167008
         weighted avg       0.99      0.99      0.99   1167008
```

Gambar 19. Perhitungan Akurasi Linear SVM

Gambar diatas menunjukan tingkat akurasi *model Support Vector Machine* sampai dengan 99%.

KESIMPULAN DAN SARAN

Berdasarkan penelitian, analisis, data dan hasil, sehingga dapat disimpulkan bahwasanya penyalahgunaan kejahatan *carding* dapat di prediksi secara akurat baik itu menggunakan Algoritma *Naïve Bayes* dan *Support vector machine*. Akan tetapi Algoritma *Support Vector Machine* lebih akurat daripada *Naïve Bayes*. Algoritma *Support Vector Machine* mendapatkan skor 99% lebih tinggi 1% daripada *Naïve Bayes*. Hal ini dikarenakan Pengklasifikasi *Support vector machine* menawarkan akurasi tinggi dan bekerja dengan baik di ruang dimensi tinggi. Pengklasifikasi SVM pada dasarnya menggunakan subset dari titik pelatihan, sehingga *output* hanya memakan sedikit memori.

Sebaiknya untuk penelitian kejahatan *carding* selanjutnya bisa dengan metode deep learning. Dengan begitu tingkat akurasi yang didapatkan jauh lebih tinggi.

ISSN 2686-3359 (Online)

DAFTAR PUSTAKA

[1] Tata Sutabri, 2012, *Komputer dan Masyarakat*, Penerbit Andi, Yogyakarta
[2] Tata Sutabri, R. Pandi Selvam, K. Shankar, Phong Thanh Nguyen, Wahidah Hashim, Andino Maseleno, 2019, *Machine Learning for Healthcare Diagnostics*, Blue Eyes Intelligence Engineering & Sciences Publication, BHOPAL, Madhya Pradesh 462021, IN
[3] Heri Syahputra, 2021, *Perancangan Sistem Pakar Untuk Mengidentifikasi Keamanan Transaksi Online Website E-commerce Dengan Menggunakan Metode Certainty Factor*, Jurnal Informasi dan Teknologi Ilmiah (INTI)
[4] Fatima Salahdine, Zakaria El Mrabet, Naima Kaabouch, 2022, *Phishing Attacks Detection A Machine Learning-Based Approach*, Cornell University, New York City.
[5] Fatima Salahdine, Zakaria El Mrabet, Naima Kaabouch, 2022, *Phishing Attacks Detection A Machine Learning-Based Approach*, Cornell University, New York City.
[6] Wildan Cika Pradana, Mochtar Yahya, Halimahtus Mukminna, 2022, *SISTEM DIAGNOSIS PENYAKIT KULIT PADA MANUSIA DENGAN METODE FORWARD CHAINING BERBASIS ANDROID*, JINTEKS, Sumbawa.
[7] Mazmur Triputra, Eri Sasmita Susanto, Wilia Ismiyarti, 2019, *RANCANG BANGUN APLIKASI KLASIFIKASI PLAGIARISME DENGAN MEMANFAATKAN MACHINE LEARNING BERBASIS ANDROID*, JINTEKS, Sumbawa.
[8] Nengah Widya Utami, 2022, *TEXT MINIG CLUSTERING UNTUK PENGELOMPOKAN TOPIK DOKUMEN PENELITIAN MENGGUNAKAN ALGORITMA K-MEANS DENGAN COSINE SIMILARITY*, JINTEKS, Sumbawa.
[9] Weifeng Li, Hsinchun Chen, Jay F. Nunamaker Jr, 2017, *Identifying and Profiling Key Sellers in Cyber Carding Community: AZSecure Text Mining System*, AZSecure Text Mining System, Journal of Management Information Systems.
[10] Chayal, N.M., Patel, N.P, 2021 , *Review of Machine Learning and Data Mining Methods to Predict Different Cyberattack*s, Springer, Singapore
[11] Hao Hua Sun Yin, Klaus Langenheldt, Mikkel Harlev, Raghava Rao Mukkamala & Ravi Vatrapu, 2019, *Regulating Cryptocurrencies: A Supervised Machine Learning Approach to De-Anonymizing the Bitcoin Blockchain*, Journal of Management Information Systems.
[12] Rami Mustafa A. Mohammad, 2022, *An Enhanced Multiclass Support Vector Machine Model and its Application to Classifying File Systems Affected by a Digital Crim*, Journal of King Saud University - Computer and Information Sciences.
[13] Xuqiao Yu, 2020, *Phishing Websites Detection Based on Hybrid Model of Deep Belief Network and Support Vector Machine*, IOP Publishing Ltd.
[14] A. V. Kachavimath, S. V. Nazare and S. S. Akki, 2020, *Distributed Denial of Service Attack Detection using Naïve Bayes and K-Nearest Neighbor for Network Forensics*, 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore.
[15] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq and M. K. Khan, 2020, *Comprehensive Review of Cybercrime Detection Techniques*, IEEE Access.