**Tugas Besar**

**ditujukan untuk memenuhi salah satu tugas mata kuliah Penambangan Data Kelas IF-41-GAB01**

**Dosen Pengampu Angelina Prima Kurniati, S.T., M.T., Ph.D.**

**Disusun oleh :**

**Luqman Haries (1301180072)**


**FAKULTAS TEKNIK INFORMATIKA**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2021**

# DAFTAR ISI

# PENDAHULUAN

## 1. Deskripsi Data

Pembayaran mandiri sudah tersedia dimana-mana, customer bisa melakukan pembayaran secara mandiri di tempat yang sudah disediakan, customer cukup melakukan scan dan membayar sejumlah produk yang dibeli. Proses pembayaran seperti ini sangat membantu memotong proses pengantrian, dan menghemat waktu, akan tetapi hal tersebut sangat rentan dari penyalahgunaan pembayaran seperti halnya penipuan kartu (*card fraud*).

Dataset *fraud* ini memiliki 2 dataset yaitu data *train* dan data *test*. Dataset tersebut berisikan informasi tentang pemindaian untuk pelanggan yang melakukan pembayaran tergolong *fraud* atau *not fraud*.

## 2. Karakteristik Data

Dataset *train* digunakan untuk membangun model yang nantinya akan digunakan sebagai prediksi dengan data *test*. Dari dataset yang diberikan terdapat atribut yang menginterpretasikan suatu kejadian yang diukur dalam bentuk *value range*, berikut adalah tabel karakteristik data beserta deskripsi yang diberikan:

| Nama Kolom | Value Range | Deskripsi |
|---|---|---|
| trustLevel | {1,2,3,4,5,6} | Tingkat kepercayaan |
| totalScanTimeInSeconds | Angka positif | Berapa lama untuk melakukan pembelian (dalam detik) |
| grandTotal | Angka desimal positif dengan 2 tempat desimal (contoh 10.99) | Berapa banyak pengeluaran ($) |

| | | |
|---|---|---|
| lineItemVoids | Angka positif | Jumlah pemindaian void |
| scansWithoutRegistration | Dari 0 hingga Angka positif | Jumlah pemindaian yang gagal |
| quantityModification | Dari 0 hingga Angka positif | Jumlah yang dimodifikasi untuk salah satu produk yang dipindai |
| scannedLineItemsPerSecond | Angka positif desimal | Jumlah rata-rata produk yang dipindai per detik |
| valuePerSecond | Angka positif desimal | Nilai total rata-rata dari produk yang dipindai per detik |
| lineItemVoidsPerPosition | Angka positif desimal | Rata-rata jumlah item voids per jumlah total semua dipindai dan tidak produk yang dibatalkan |
| fraud | {0,1} | Klasifikasi dengan 1 adalah *fraud* dan 0 adalah *not fraud* |

**3. Tujuan Data**

Tujuan dataset untuk membuat model klasifikasi pemindaian penggunaan kartu tersebut tergolong *fraud* atau *non-fraud*. Klasifikasi tidak memperhitungkan pengguna *fraud* tersebut dengan sengaja melakukan *fraud* dengan sengaja atau tidak.

## PREPROSES

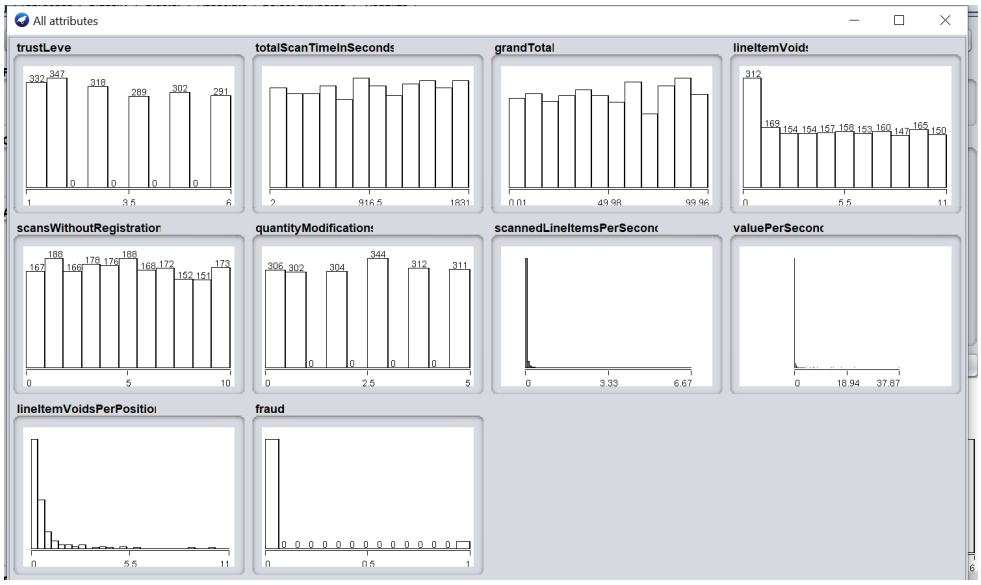
### 1. Data Cleaning

Dataset yang digunakan akan dilakukan preproses untuk melihat atribut yang memiliki masalah pada data. Masalah - masalah data yang akan dilihat pada dataset *train* tersebut antara lain adalah *missing value,* dan *noise data*.

#### *a. Missing Value*

*Missing value* adalah data yang hilang pada pada dataset, masalah tersebut akan sangat berpengaruh terhadap hasil yang akan dikeluarkan ketika jumlah data yang hilang sangat besar. *Missing value* dapat menyebabkan tingkat keakuratan suatu informasi menjadi kurang dan bahkan tidak akan mendapatkan informasi yang berguna. Oleh karena itu, pada dataset *train* akan dilakukan pengecekan nilai kolom yang memiliki *missing value*.

Pada dataset yang digunakan tidak terdapat masalah *missing value*, berikut adalah visualisasinya:



### *b. Noise data*

*Noise data* adalah data yang nilainya diluar dari penetapan pada karakteristik data atau data yang berbeda dari data lainnya sehingga data tersebut tidak memiliki informasi yang penting atau data yang rusak. *Noise data* mencangkup data apapun yang tidak dapat dipahami dan ditafsirkan. Masalah tersebut dapat terjadi karena kesalahan sistem ataupun kesalahan faktor manusia seperti kesalahan input.

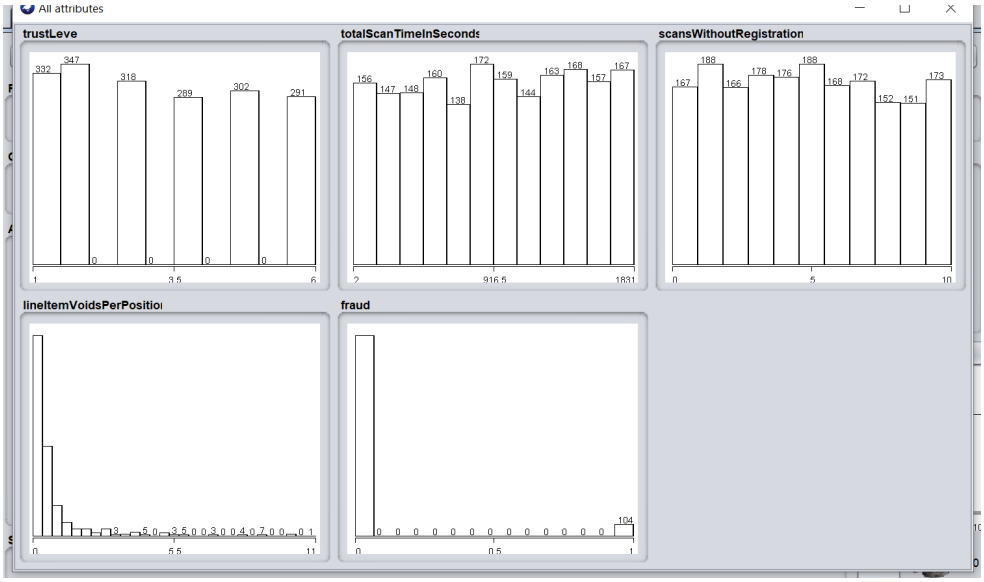
Pada dataset yang digunakan tidak terdapat masalah *noise data*, karena *value* yang berada pada masing masing kolom tidak melebihi yang ada pada *value range* pada tabel karakteristik data, dapat dilihat pada informasi *min* dan *max*.

| | trustLevel | totalScanTimeInSeconds | grandTotal | lineItemVoids | scansWithoutRegistration | quantityModifications | scannedLineItemsPerSecond | valuePerSecond | lineItemVoidsPerPosition | fraud |
|---|---|---|---|---|---|---|---|---|---|---|
| count | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 | 1879.000000 |
| mean | 3.401809 | 932.153273 | 50.864492 | 5.469931 | 4.904204 | 2.525279 | 0.058138 | 0.201746 | 0.745404 | 0.055349 |
| std | 1.709404 | 530.144640 | 28.940202 | 3.451169 | 3.139697 | 1.695472 | 0.278512 | 1.242135 | 1.327241 | 0.228720 |
| min | 1.000000 | 2.000000 | 0.010000 | 0.000000 | 0.000000 | 0.000000 | 0.000548 | 0.000007 | 0.000000 | 0.000000 |
| 25% | 2.000000 | 474.500000 | 25.965000 | 2.000000 | 2.000000 | 1.000000 | 0.008384 | 0.027787 | 0.160000 | 0.000000 |
| 50% | 3.000000 | 932.000000 | 51.210000 | 5.000000 | 5.000000 | 3.000000 | 0.016317 | 0.054498 | 0.350000 | 0.000000 |
| 75% | 5.000000 | 1397.000000 | 77.285000 | 8.000000 | 8.000000 | 4.000000 | 0.032594 | 0.107313 | 0.666667 | 0.000000 |
| max | 6.000000 | 1831.000000 | 99.960000 | 11.000000 | 10.000000 | 5.000000 | 6.666667 | 37.870000 | 11.000000 | 1.000000 |

## 2. Attribute Selection

Dataset yang digunakan memiliki jumlah atribut yang sangat besar pada *train* maupun *test*. Dari atribut yang sangat besar tersebut akan dianalisis keterkaitan atribut tersebut dengan *class fraud*. Tujuan dari analisis keterkaitan ini untuk menemukan sekumpulan kolom yang relevan untuk proses prediksi. Kolom yang tidak relevan akan dihapus karena dinilai tidak banyak mempengaruhi hasil prediksi dan mempercepat proses prediksi karena melakukan reduksi data.

Attribute yang dinilai penting dalam pembentukan model didapat menggunakan *Weka* dengan menggunakan filter *Attribute Selection*, dapat dilihat dari gambar berikut:



# MODELING

## 1. Metode

Metode klasifikasi yang akan dilakukan pada dataset *fraud* ini akan menggunakan 3 metode yaitu *knn classification*, *naive bayes* dan *decision tree*. *Knn* adalah algoritma yang hasil klasifikasinya memperhitungkan berdasarkan jarak kedekatan antara objek. *Naive bayes* metode statistik yang mengasumsikan semua data saling berkorelasi. *Decision tree* algoritma yang struktur perhitungannya menyerupai struktur pohon.

## 2. Tahap yang dilakukan

### a. Pembangunan Model

```
[9]  #getting feature importance from corr to classification to X and label to Y
     feature_names = ['trustLevel','totalScanTimeInSeconds','scansWithoutRegistration','lineItemVoidsPerPosition']

[10] #getting train and test to variable
     X_train = df_train[feature_names]
     y_train = df_train['fraud']

     X_test = df_test[feature_names]
     y_test = df_test['fraud']

[11] #Apply scalling
     from sklearn.preprocessing import MinMaxScaler
     scaler = MinMaxScaler()
     X_train = scaler.fit_transform(X_train)
     X_test = scaler.transform(X_test)
```

Atribut yang digunakan untuk membangun model didasari dari hasil *Attribute selection* karena atribut-atribut yang muncul adalah atribut yang memiliki korelasi dengan class *fraud*. Kemudian melakukan *scaling* pada dataset *train* dan *test* yang berguna untuk menyesuaikan rentang atau *range* pada data.

### b. Evaluation dengan F1-Score

*F-measure* atau *F1-score* merupakan perhitungan evaluasi dalam informasi yang mengkombinasikan recall dan precision. Nilai yang dihasilkan oleh metode ini adalah 0 hingga 1, dimana nilai 0 menandakan model tersebut

tidak dapat menghasilkan prediksi yang akurat, dan nilai 1 menandakan model tersebut menghasilkan prediksi yang akurat. Metode ini digunakan untuk keadaan dimana bobot dari data *train* dan *test* memiliki rentang *instance* yang jauh berbeda atau *unbalance*, karena *instance* pada data *train* sebanyak 1900 dan pada data *test* sebanyak 500 000.

## 3. Hasil

Hasil Akurasi dari setiap algoritma *knn, naive bayes,* dan *decision tree*

| **Algoritma** | **Akurasi** |
|---|---|
| *Knn classification* | 94.013 % |
| *Naive bayes* | 92.104 % |
| *Decision Tree* | 94.391 % |

Hasil *F1-Score* dari setiap algoritma *knn, naive bayes,* dan *decision tree*

| **Algoritma** | **Score** |
|---|---|
| *Knn classification* | 0.9401269972556868 |
| *Naive bayes* | 0.9210392655599744 |
| *Decision Tree* | 0.9439051957255365 |

# KESIMPULAN

## 1. Interpretasi

Dari dataset *fraud* data mining ini menghasilkan perbandingan menggunakan 3 model algoritma, *knn classification, naive bayes,* dan *decision tree*, dengan menggunakan model didapat menghasilkan nilai akurasi sebesar 94.391% untuk algoritma *Decision tree*, kemudian *knn* dengan akurasi sebesar 94.013 %, dan *naive bayes* dengan akurasi sebesar 92.104 % dan dari perbandingan *F1-score* nilai yang mendekati 1 adalah algoritma *decision tree,* jadi dapat disimpulkan dataset *fraud*, yang terbaik menggunakan model *Decision tree.*

## 2. Dokumentasi

Google Colab:
https://colab.research.google.com/drive/15CgzR1BfKkPBLiS5C1xGO52hcS3dnK2q?usp=sharing

Video Presentasi:
https://drive.google.com/file/d/1P4ZMW2qtAzFKbn0g_ZjUpedYxx8JQXrF/view?usp=sharing

Slide Presentasi:
https://docs.google.com/presentation/d/1DP_Fp-SWGGV2vuDXxbdQNy_EfXEJfZIIwrtSuELDeq4/edit?usp=sharing

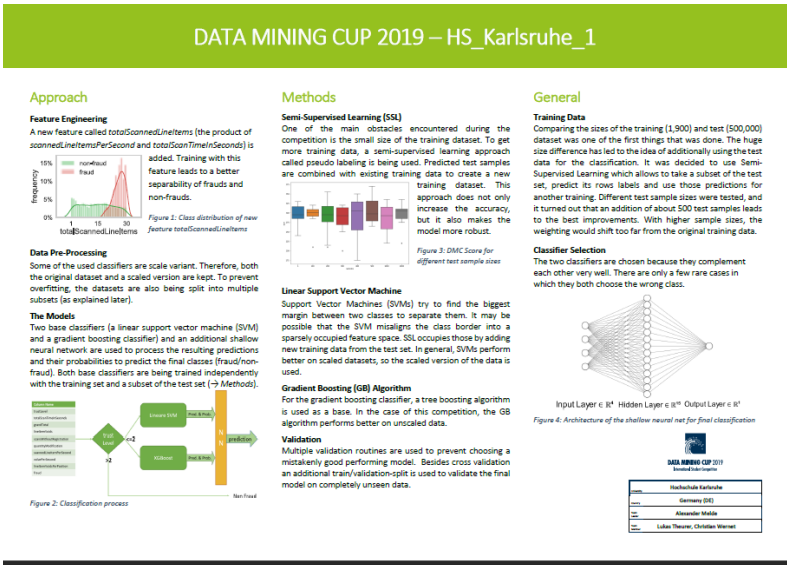
**REFERENSI**


Attribute Subset Selection in Data Mining - GeeksforGeeks

DMC_2019/var_importance.pdf at master · kozodoi/DMC_2019 (github.com)

Comparative Study on Classic Machine learning Algorithms | by Danny Varghese | Towards Data Science

10.1007/978-3-319-78503-5_6.pdf (springer.com)

DATA MINING CUP 2019 – HS_Karlsruhe_1

Approach

Feature Engineering

A new feature called *totalScannedLineItems* (the product of *scannedLineItemsPerSecond* and *totalScanTimeInSeconds*) is added. Training with this feature leads to a better separability of frauds and non-frauds.

Figure 1: Class distribution of new feature totalScannedLineItems

Data Pre-Processing

Some of the used classifiers are scale variant. Therefore, both the original dataset and a scaled version are kept. To prevent overfitting, the datasets are also being split into multiple subsets (as explained later).

The Models

Two base classifiers (a linear support vector machine (SVM) and a gradient boosting classifier) and an additional shallow neural network are used to process the resulting predictions and their probabilities to predict the final classes (fraud/non-fraud). Both base classifiers are being trained independently with the training set and a subset of the test set (→ *Methods*).

Figure 2: Classification process

Methods

Semi-Supervised Learning (SSL)

One of the main obstacles encountered during the competition is the small size of the training dataset. To get more training data, a semi-supervised learning approach called pseudo labeling is being used. Predicted test samples are combined with existing training data to create a new training dataset. This approach does not only increase the accuracy, but it also makes the model more robust.

Figure 3: DMC Score for different test sample sizes

Linear Support Vector Machine

Support Vector Machines (SVMs) try to find the biggest margin between two classes to separate them. It may be possible that the SVM misaligns the class border into a sparsely occupied feature space. SSL occupies those by adding new training data from the test set. In general, SVMs perform better on scaled datasets, so the scaled version of the data is used.

Gradient Boosting (GB) Algorithm

For the gradient boosting classifier, a tree boosting algorithm is used as a base. In the case of this competition, the GB algorithm performs better on unscaled data.

Validation

Multiple validation routines are used to prevent choosing a mistakenly good performing model. Besides cross validation an additional train/validation-split is used to validate the final model on completely unseen data.

General

Training Data

Comparing the sizes of the training (1,900) and test (500,000) dataset was one of the first things that was done. The huge size difference has led to the idea of additionally using the test data for the classification. It was decided to use Semi-Supervised Learning which allows to take a subset of the test set, predict its rows labels and use those predictions for another training. Different test sample sizes were tested, and it turned out that an addition of about 500 test samples leads to the best improvements. With higher sample sizes, the weighting would shift too far from the original training data.

Classifier Selection

The two classifiers are chosen because they complement each other very well. There are only a few rare cases in which they both choose the wrong class.

Input Layer $\in \mathbb{R}^4$ Hidden Layer $\in \mathbb{R}^{15}$ Output Layer $\in \mathbb{R}^1$

Figure 4: Architecture of the shallow neural net for final classification

DATA MINING CUP 2019

| | |
|---|---|
| University | Hochschule Karlsruhe |
| Country | Germany (DE) |
| Team Member | Alexander Malde |
| Team Member | Lukas Theurer, Christian Wiernet |