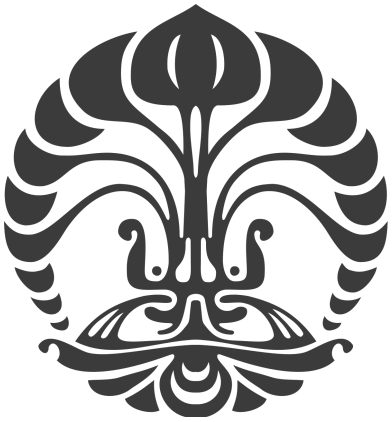
MAKALAH
PENDETEKSI KELAYAKAN PENGAJUAN KREDIT OLEH DEBITUR DENGAN MENGGUNAKAN TEKNOLOGI MACHINE LEARNING



Untuk Mata Kuliah Sains Data – SCMA602017

Penyusun:

Annisa Zahra	2006463295	2020
Hosia Josindra Saragih	2006463332	2020
Angelica Patricia Djaya Saputra	2006522000	2020

DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS INDONESIA
DEPOK

2022

ABSTRAK

Penilaian kredit yang membantu dalam mengevaluasi kemampuan pembayaran kembali dari calon peminjam merupakan salah satu masalah terpenting bagi lembaga pinjaman. Pada saat ini, perkembangan *Machine Learning* sangat pesat dan penggunaannya dapat diimplementasikan dalam berbagai hal, salah satunya dalam lembaga pinjaman. Pada penelitian ini, peneliti menggunakan model *Decision Tree* untuk menentukan debitur yang tidak membayar kembali pinjamannya. Studi ini akan membandingkan kinerja ketiga model antara lain *Decision Tree*, *Logistic Regression*, dan *Support Vector Machine*. Hasilnya menunjukkan bahwa model *Decision Tree* bekerja jauh lebih baik.

Kata kunci: Kredit, Debitur, *Machine Learning*, *Decision Tree*, *Logistic Regression*, *Support Vector Machine*

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Berdasarkan riset, pengajuan kredit oleh debitur terdeteksi telah meningkat. Hal ini mampu menjadi peluang bagi kreditur untuk mendapatkan keuntungan. Namun, tidak semua debitur mampu membayar pengajuan kreditnya tepat waktu atau bahkan tidak dibayar sama sekali. Hal ini mampu merugikan kreditur. Sekarang tantangan yang dihadapi adalah bagaimana untuk mengoptimalkan penerimaan pengajuan kredit?

Machine Learning merupakan mesin yang mampu belajar dengan sendirinya layaknya seorang manusia yang belajar dari masa lalu. *Machine Learning* dapat diimplementasikan dalam berbagai hal, salah satunya adalah bidang keuangan. Disini, akan dibandingkan tiga model dari *Machine Learning* sebagai pendeteksi bertipe klasifikasi yaitu *Logistic Regression, Decision Tree*, dan *Support Vector Machine*. Kemudian akan dilakukan perbandingan terhadap ketiga model untuk mendapatkan model terbaik.

1.2 Rumusan Masalah

1. Apa saja faktor yang dapat mendeteksi debitur yang berisiko?
2. Bagaimana penerapan *Machine Learning* dalam mendeteksi debitur yang berisiko?
3. Apakah model yang terbaik dalam mendeteksi debitur yang berisiko pada dataset yang digunakan?

1.3 Tujuan Penelitian

1. Untuk mengetahui faktor-faktor yang dapat mendeteksi debitur yang berisiko
2. Untuk mengetahui penerapan *Machine Learning* dalam mendeteksi debitur yang berisiko
3. Untuk mengetahui model yang terbaik dalam mendeteksi debitur yang berisiko pada dataset yang digunakan

BAB II
DATA DAN METODE

2.1 Data

Peneliti menggunakan dataset berisi kolom yang mensimulasikan data biro kredit. Peneliti mendapatkan dataset ini berasal dari kaggle. Isi dari dataset yang dipilih peneliti adalah sebagai berikut,

a. person_age : umur
b. person_income : pendapatan tahunan
c. person_home_ownership: kepemilikan rumah
d. person_emp_length: lama kerja (dalam tahun)
e. loan_intent: niat pinjaman
f. loan_grade: kelas pinjaman (A (paling tidak berisiko) - G (paling berisiko))
g. loan_amnt: jumlah pinjaman
h. loan_int_rate: suku bunga
i. loan_status: status pinjaman (non default: 0, default: 1)
j. loan_percent_income: persen pendapatan
k. cb_person_default_on_file: histori default
l. cb_person_cred_hist_length: panjang riwayat kredit

Dalam dataset tersebut, terdapat 12 kolom dengan 5 kolom data integer, 3 kolom data float, dan 4 kolom data object yang berukuran 32581 baris x 12 kolom.

Variabel target dalam dataset tersebut adalah fitur loan_status yang dapat menentukan kreditur tersebut berisiko atau tidak. Jika kreditur mendapatkan nilai 0 (non default) maka kreditur tersebut tidak berisiko sedangkan nilai 1 (default) menandakan kreditur tersebut berisiko.

2.2 Metode

Akan digunakan 3 (tiga) metode pada *Machine Learning* sebagai model bertipe klasifikasi.

2.2.1 *Logistic Regression*

Logistic Regression menggunakan konsep regresi untuk memprediksi suatu nilai yang bersifat kategorik. Metode ini akan memberikan output berupa probabilitas suatu item termasuk kelas tertentu.

2.2.2 *Decision Tree*

Decision Tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki.

2.2.3 *Support Vector Machine*

Support Vector Machine (SVM) adalah algoritma *supervised learning* untuk klasifikasi dengan cara menemukan separator berupa *hyperplane*.

BAB III
IMPLEMENTASI DAN ANALISIS DATA

3.1 Implementasi

3.1.1 Cleaning Data

Sebelum menggunakan dataset, peneliti akan melakukan cleaning data untuk menangani *missing values* dan *outliers* pada dataset. Setelah dilakukan *cleaning data*, didapatkan *missing values* dan *outliers* pada dataset. *Missing values* terdapat pada kolom person_emp_length sebanyak 895 baris dan kolom loan_int_rate sebanyak 3115 baris. Hal tersebut akan ditangani dengan memasukkan nilai rata-rata ke kolom tersebut. Kami dapat mendeteksi *outliers* dengan membuat boxplot dan didapatkan *outliers* pada kolom person_age dan kolom person_emp_length. Hal tersebut kami tangani dengan menghapus data person_age>100 tahun dan menghapus data person_emp_length>person_age.

3.1.2 Encoding

Encoding adalah salah satu tahap praproses data sebelum data diproses dengan algoritma machine learning. Dalam dataset pasti akan dijumpai beberapa fitur yang bertipe kategori seperti di dataset kami adalah person_home_ownership, loan_intent, loan_grade, dan cb_person_default_on_file yang didapat dengan mencari data yang bertipe object.

Dari data yang bertipe kategorik tersebut didapat 2 jenis data yaitu ordinal (tingkatan) dan nominal (diskrit). Kami melakukan One-hot encoding untuk kolom-kolom bertipe nominal yaitu person_home_ownership, loan_intent, dan cb_person_default_on_file dan dilakukan Ordinal encoding untuk kolom bertipe ordinal yaitu ‘loan_grade’.

3.1.3 Uji Multikolinearitas

Multikolinearitas merupakan keadaan dimana terdapat korelasi yang kuat antara dua atau lebih variabel bebas dalam model regresi. Oleh karena itu, peneliti menguji multikolinearitas pada variabel bebas pada data yang telah diproses dengan menggunakan modul *VIF (Variance Inflation Factor)*. Kemudian didapatkan beberapa feature memiliki *VIF’s score* tak hingga yang mengartikan bahwa terdapat korelasi sempurna pada di antara variabel yang ada sehingga variabel tersebut redundant. Akan tetapi, kasus ini terjadi karna fitur yang memiliki *VIF’s score* tak hingga merupakan fitur hasil dari *OneHot encoding*, sehingga memiliki korelasi yang sempurna maka fitur-fitur tersebut tetap akan digunakan.

Lalu, untuk *feature* lainnya memiliki *VIF's score* kurang dari 10 sehingga tidak terjadi multikolinearitas. Maka seluruh *feature* siap untuk tahap berikutnya yakni *feature selection.*

3.1.4 Feature Selection

Sebelum menggunakan *Machine Learning*, peneliti akan melakukan proses *feature selection* untuk dapat menentukan fitur-fitur yang paling berpengaruh dalam menentukan apakah debitur berisiko. Sebelum melakukan *feature selection*, kami melakukan *encoding* pada kolom yang masih bertipe object ke tipe numerik. Dalam feature selection, kami menggunakan uji multikolinearitas dengan *Variance Inflation Factor* (VIF) dan *Pearson Correlation*.

Dari uji multikolinearitas dengan VIF didapatkan bahwa tidak ada fitur yang multikolinearitas (memberi dampak yang sama) sehingga tidak ada fitur yang dihapus. Kemudian, pada *Pearson Correlation* didapatkan bahwa terdapat delapan fitur yang berpengaruh dalam menentukan debitur yang berisiko adalah person_income, loan_grade, loan_int_rate, loan_percent_income, person_home_ownership_MORTGAGE, person_home_ownership_RENT, cb_person_default_on_file_N, dan cb_person_default_on_file_Y.

3.1.5 Splitting dan Normalisasi Data

Kami melakukan train/test split yaitu metode untuk mengevaluasi performa model *Machine Learning* dengan membagi dataset menjadi dua bagian yaitu untuk training data dan testing data dengan proporsi pembagiannya adalah 8:2.

Selanjutnya, kami melakukan normalisasi data yaitu proses mengubah data numerik memiliki rentang nilai yang sama yaitu dari 0 sampai 1 dengan MinMaxScaler.

3.1.6 Tuning Hyperparameter

Tuning Hyperparameter dilakukan untuk melakukan tuning pencarian parameter terbaik untuk ketiga model dengan menggunakan GridSearchCV. Didapatkan hasil sebagai berikut,

1. Untuk model *Decision Tree*, didapatkan akurasi sebesar 0.894969 dengan parameter terbaiknya adalah 'criterion': 'gini', 'max_depth': 8, 'max_features': 'log2', 'splitter': 'best'.
2. Untuk model *Logistic Regression*, didapatkan akurasi sebesar 0.850263 dengan parameter terbaiknya adalah 'C': 0.1, 'multi_class': 'multinomial', 'penalty': 'none', 'solver': 'sag'.

3. Untuk model SVM, didapatkan akurasi sebesar 0.872558 dengan parameter terbaiknya adalah 'gamma': 'scale', 'kernel': 'poly'.

3.1.7 Pembuatan Model

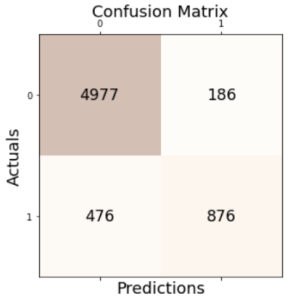
Melakukan pembentukan model untuk klasifikasi data dengan 3 (tiga) metode yaitu Logistic Regression, Decision Tree, dan Support Vector Machine dengan bantuan modul sklearn.

3.2 Analisis Data

3.2.1 Evaluasi Model

Ketiga model yang telah peneliti bentuk sebelumnya, akan dievaluasi menggunakan modul *classification report*. Dalam evaluasi model ini kita akan lebih berfokus kepada nilai *f1-score* terbesar, karena berdasarkan formulanya *f1-score* mengandung *precision* dan *recall*. Pada tahap ini diperoleh *f1-score* terbesar yaitu 0.94 pada model yang memanfaatkan metode *decision tree*, maka peneliti memilih model terbaik untuk dievaluasi lebih lanjut yaitu model dengan metode *decision tree*.

3.2.1.1 *Confusion Matrix*



Gambar 1. *Confusion Matrix Decision Tree Model*

Diperoleh analisis evaluasi model (berdasarkan *data test*) sebagai berikut:

1. *True Negative*: model memprediksi 4977 debitur layak menerima kredit dan memang benar layak.
2. *False Negative*: model memprediksi 476 debitur layak menerima kredit padahal seharusnya tidak layak.
3. *True Positive*: model memprediksi 876 debitur tidak layak menerima kredit dan memang benar tidak layak.

4. *False Positive*: model memprediksi 186 debitur tidak layak menerima kredit padahal seharusnya layak.

3.2.1.2 *Classification Report*

Setelah memperoleh *true negative, false negative, true positive,* dan *false positive* maka didapatkan *classification report* dari model sebagai berikut.

Classification Report Decision Tree Model

	precision	recall	f1-score	support
0	0.91	0.96	0.94	5163
1	0.82	0.65	0.73	1352
accuracy			0.90	6515
macro avg	0.87	0.81	0.83	6515
weighted avg	0.89	0.90	0.89	6515

f1-score: 0.9376412961567445

Gambar 2. *Classification Report Decision Tree Model*

Berdasarkan *classification report* diperoleh untuk kelas 0 (pengajuan kredit diterima):

1. *Precision*: 0.91
2. *Recall*: 0.96
3. *F1-score*: 0.94
4. *Support* (jumlah *data test* kelas 0): 5163

Selanjutnya, untuk kelas 1 (pengajuan kredit ditolak) diperoleh:

1. *Precision*: 0.82
2. *Recall*: 0.65
3. *F1-score*: 0.73
4. *Support* (jumlah *data test* kelas 1): 1352

disimpulkan model terbaik menggunakan metode *decision tree* dengan *f1-score* sebesar 0.94 .

3.2.2 Implementasi Model (Aplikasi)

Kredita merupakan aplikasi pendeteksi kelayakan debitur, dengan memfokuskan pada *feature-feature* terpilih sebelumnya yakni pendapat tahunan, kelas pinjamin, suku bunga, persen pendapatan, kepemilikan rumah gadai/sewa, riwayat gagal bayar.

```
Credit Risk Prediction Programming Algorithm
Person Income: 120000
Loan Grade: A
Loan Interest Rate: 7.49
Loan Percent Income: 0.15
Person Home Ownership: mortgage
CB Person Default on File: n
Kelayakan Penerima Nasabah Kredit:
Layak
```

```
Credit Risk Prediction Programming Algorithm
Person Income: 76000
Loan Grade: B
Loan Interest Rate: 10.99
Loan Percent Income: 0.46
Person Home Ownership: RENT
CB Person Default on File: N
Kelayakan Penerima Nasabah Kredit:
Tidak Layak
```

Gambar 3. *Algoritma Program (Pyhton)*

Gambar 4. *Implementasi Dalam Bentuk Aplikasi*

Sehingga dengan dibentuknya implementasi dari model dalam sebuah aplikasi, pengujian kelayakan debitur dalam menerima kredit akan lebih praktis dan efektif.

BAB IV
KESIMPULAN

Berdasarkan *Classification Report*, diperoleh F1-score tertinggi didapat pada model *decision tree* sebesar 0.94 atau 94% yang hasilnya mendekati 1 sehingga menunjukan bahwa performa model sangat baik. Model ini menggunakan parameter berupa 'criterion': 'gini','max_depth': 8,'max_features': 'log2','splitter': 'best’.

Interpretasi dari hasil yang peneliti peroleh memiliki kesimpulan bahwa untuk mengetahui debitur layak menerima kredit atau tidak dengan memperhatikan jumlah pendapat tahunan, kelas pinjamin, suku bunga, persen pendapatan, kepemilikan rumah gadai/sewa, riwayat gagal bayar. Pada aplikasi yang dibentuk, ketika dilakukan submit, akan muncul *approved* atau *rejected* sesuai dengan tingkat kelayakan penerima nasabah kredit.

REFERENSI

Team, T. I. (2022, March 17). What is credit risk? Investopedia. Retrieved June 12, 2022, from https://www.investopedia.com/terms/c/creditrisk.asp

Tse, L. (2020, June 2). Credit risk dataset. Kaggle. Retrieved June 12, 2022, from https://www.kaggle.com/datasets/laotse/credit-risk-dataset?resource=download

Tuovila, A. (2021, June 17). Detection risk definition. Investopedia. Retrieved June 12, 2022, from https://www.investopedia.com/terms/d/detection-risk.asp