

Sistem Pengambilan Keputusan Penentuan Kelayakan Penerima Kredit Menggunakan K-Nearest Neighbors dan Ensemble Bagging

Dosen Pengampu:
Yahya Nur Ifriza, S.Pd., M.Kom.



Kelompok 03:

Muhammad Veka Syahputra	:	4612422022
Zamachsyafi Shidqi Athallah	:	4612422025
Fauzan Dwi Eryawan	:	4612422037
Muhammad Rizqi Rahman	:	4612422028
Agus Sutiyanto	:	4612422052

**Prodi Sistem Informasi
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Semarang
Tahun 2023**

K-Nearest Neighbors

K-Nearest Neighbors (KNN) adalah algoritma machine learning yang berfungsi untuk melakukan klasifikasi suatu data berdasarkan data pembelajaran (train data sets), yang diambil dari k tetangga terdekatnya (nearest neighbors). Dengan k merupakan banyaknya tetangga terdekat. Adapun dalam proyek kali ini menggunakan K-Nearest Neighbors (KNN) untuk melakukan klasifikasi kelayakan pengajuan kredit, klasifikasi KNN dilakukan sebagai pendukung keputusan untuk menerima apakah konsumen layak menerima kredit berdasarkan kriteria tertentu. Hal tersebut diperlukan untuk menghindari kredit macet atau konsumen tidak mampu membayar tagihan kredit, sehingga diperlukan sebuah klasifikasi untuk menentukan kelayakan kredit. Berikut adalah kriteria yang digunakan untuk menilai permohonan kredit

1. Tabel Pekerjaan

No	Pekerjaan	Nilai	Bobot
1	Petani	1	0.10
2	Wiraswasta dan Lain-lain	2	0.15
3	Karyawan	3	0.20
4	Profesi	4	0.25
5	PNS/BUMN	5	0.30

2. Tanggungan

No	Tanggungan	Nilai	Bobot
1	>6 Orang	1	0.10
2	5 Orang	2	0.15
3	3-4 Orang	3	0.20
4	1-2 Orang	4	0.25
5	0	5	0.30

3. Penghasilan

No	Penghasilan	Nilai	Bobot
1	< 1 Juta Rupiah	1	0.10
2	1 s/d 1,5 Juta Rupiah	2	0.15
3	1,5 s/d 2,5 Juta Rupiah	3	0.20
4	2,5 s/d 3,5 Juta Rupiah	4	0.25
5	> 3,5 Juta Rupiah	5	0.30

4. Status Rumah

No	Status Rumah	Nilai	Bobot
1	Kos/Kontrak	1	0.10
2	KPR	2	0.15
3	Milik Instansi	3	0.20
4	Milik Keluarga	4	0.25
5	Milik Sendiri	5	0.30

5. Kelengkapan Berkas

No	Kelengkapan Berkas	Nilai	Bobot
1	Sangat Kurang	1	0.10
2	Kurang	2	0.15
3	Cukup	3	0.20
4	Baik	4	0.25
5	Sangat Baik	5	0.30

Perhitungan Menggunakan KNN

Dalam melakukan proses mencari jarak tetangga terdekat dalam KNN, maka yang paling umum digunakan adalah euclidean distance, yang menghitung jarak antara titik data baru dengan data training. Berikut adalah rumus euclidean distance.

Rumus Euclidean Distance

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

$$d(p, q) = \sqrt{w_1(p_1 - q_1)^2 + w_2(p_2 - q_2)^2 + \dots + w_n(p_n - q_n)^2}$$

Keterangan

d = Jarak

w_i = bobot

p_i = data testing atau inputan pengguna

q_i = data train dari dataset

Data Testing

Kriteria	Nilai
PNS/BUMN	5
3-4 Orang	3
2,5 s/d 3,5 Juta Rupiah	4
Milik Instansi	3
Baik	4
Hasil	Dicari dengan KNN

Menghitung jarak Euclidean Distance masing masing kriteria terhadap data training, dalam hal ini nilai k ditetapkan dengan 9, yang berarti akan mengurutkan 9 tetangga terdekat dan mengambil keputusan dari mayoritas kelas tetangga tersebut.

$$d1 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2}$$

$$d1 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2}$$

$$d1 = \sqrt{0}$$

$$d1 = 0$$

$$d2 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 2)^2 + 0.25 \times (4 - 4)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2}$$

$$d2 = \sqrt{0.30 \times (0)^2 + 0.20 \times (1)^2 + 0.25 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2}$$

$$d2 = \sqrt{0.20 \times (1)^2} = 0.20$$

$$d2 = 0,447214$$

$$d3 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2 + 0.20 \times (3 - 2)^2 + 0.25 \times (4 - 4)^2}$$

$$d3 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2 + 0.20 \times (1)^2 + 0.25 \times (0)^2}$$

$$d3 = \sqrt{0.20 \times (1)^2} = 0.20$$

$$d3 = 0,447214$$

$$d4 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2 + 0.20 \times (3 - 4)^2 + 0.25 \times (4 - 4)^2}$$

$$d4 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2 + 0.20 \times (-1)^2 + 0.25 \times (0)^2}$$

$$d4 = \sqrt{0.20 \times (-1)^2} = 0.20$$

$$d4 = 0,447214$$

$$d5 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 4)^2 + 0.25 \times (4 - 4)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2}$$

$$d5 = \sqrt{0.30 \times (0)^2 + 0.20 \times (-1)^2 + 0.25 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2}$$

$$d5 = \sqrt{0.20 \times (-1)^2} = 0.20$$

$$d5 = 0,447214$$

$$d6 = \sqrt{0.30 \times (5 - 5)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 3)^2 + 0.20 \times (3 - 3)^2 + 0.25 \times (4 - 4)^2}$$

$$d6 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (1)^2 + 0.20 \times (0)^2 + 0.25 \times (0)^2}$$

$$d6 = \sqrt{0.25 \times (1)^2} = 0.25$$

$$d6 = 0.50$$

$$d7 = \sqrt{0.30 \times (5-5)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-4)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-3)^2}$$

$$d7 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (1)^2 + 0.20 \times (0)^2 + 0.25 \times (1)^2}$$

$$d7 = \sqrt{0.25 \times (1)^2} = 0.25$$

$$d7 = 0.50$$

$$d8 = \sqrt{0.30 \times (5-5)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-4)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-5)^2}$$

$$d8 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (1)^2 + 0.20 \times (0)^2 + 0.25 \times (-1)^2}$$

$$d8 = \sqrt{0.25 \times (-1)^2} = 0.25$$

$$d8 = 0.50$$

$$d9 = \sqrt{0.30 \times (5-5)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-5)^2 + 0.20 \times (3-3)^2 + 0.25 \times (4-4)^2}$$

$$d9 = \sqrt{0.30 \times (0)^2 + 0.20 \times (0)^2 + 0.25 \times (-1)^2 + 0.20 \times (0)^2 + 0.25 \times (-1)^2}$$

$$d9 = \sqrt{0.25 \times (-1)^2} = 0.25$$

$$d9 = 0.50$$

➡ Masukkan nilai untuk K1: 5
 Masukkan nilai untuk K2: 3
 Masukkan nilai untuk K3: 4
 Masukkan nilai untuk K4: 3
 Masukkan nilai untuk K5: 4
 Probabilitas diterima: 88.89%
 Memenuhi Kelayakan

Hasil Kedekatan 9 Tetangga Terdekat (Diurutkan Berdasarkan Jarak):

	K1	K2	K3	K4	K5	Hasil	Jarak
0	5	3	4	3	4	1	0.000000
1	5	2	4	3	4	1	0.447214
2	5	3	4	2	4	0	0.447214
3	5	3	4	4	4	1	0.447214
4	5	4	4	3	4	1	0.447214
5	5	3	3	3	4	1	0.500000
6	5	3	4	3	3	1	0.500000
7	5	3	4	3	5	1	0.500000
8	5	3	5	3	4	1	0.500000

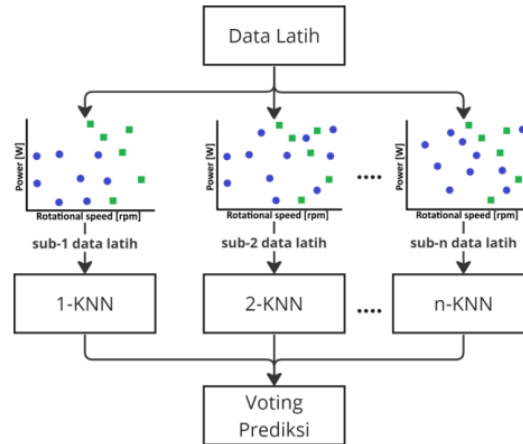
Karena mayoritas dari 9 tetangga terdekat memiliki kelas/label dengan nilai 1 maka dapat disimpulkan bahwa **Memenuhi Kelayakan**

Link Colab : <https://colab.research.google.com/drive/1Rq7njvsumvz-Uj8xXbPbe9Vtf8MxTo3Y?usp=sharing>

Ensemble Bagging

Bagging (Bootstrap Aggregating) adalah sebuah metode ensemble learning yang digunakan untuk meningkatkan kinerja model prediksi dengan menggabungkan hasil dari beberapa model yang dibuat dengan menggunakan dataset yang sama namun diambil secara acak dengan penggantian (bootstrap). Setiap model yang dihasilkan dari bagging biasanya dibangun menggunakan algoritma pembelajaran yang sama, namun dengan variasi dataset yang digunakan untuk melatihnya. Fungsi utama dari bagging adalah untuk mengurangi varians model. Dengan membuat

beberapa model yang berbeda berdasarkan subset dataset yang berbeda, bagging memungkinkan untuk mengurangi efek overfitting pada model tunggal dan menghasilkan prediksi yang lebih konsisten dan akurat.



Implementasi Menggunakan Scikit Learn

```
# Bagging
from sklearn.ensemble import BaggingClassifier
bagging_classifier = BaggingClassifier(estimator=KNeighborsClassifier(n_neighbors=9), n_estimators=10, random_state=42)

bagging_classifier.fit(X_train, y_train)

X_test = dataset_test.iloc[:, :-1]
y_test = dataset_test.iloc[:, -1]

predictions = bagging_classifier.predict(X_test)

# akurasi
accuracy = accuracy_score(y_test, predictions)
print("\nAkurasi", accuracy)

# confusion matrix
conf_matrix = confusion_matrix(y_test, predictions)
print("\nConfusion matrix:\n", conf_matrix)
```

Jadi dilakukan metode bagging pada model KNN untuk menghasilkan prediksi yang konsisten dan akurat, setelah itu print akurasi dan dan confusion matrix untuk melihat seberapa baik model knn dalam melakukan prediksi yang layak dan tidak layak untuk menerima kredit.

Hasil Akurasi Model KNN

```
➡ Masukkan nilai untuk K1: 5
Masukkan nilai untuk K2: 3
Masukkan nilai untuk K3: 4
Masukkan nilai untuk K4: 3
Masukkan nilai untuk K5: 4
Probabilitas diterima: 88.89%
Memenuhi Kelayakan
```

Hasil Kedekatan 9 Tetangga Terdekat (Diurutkan Berdasarkan Jarak):

	K1	K2	K3	K4	K5	Hasil	Jarak
0	5	3	4	3	4	1	0.000000
1	5	2	4	3	4	1	0.447214
2	5	3	4	2	4	0	0.447214
3	5	3	4	4	4	1	0.447214
4	5	4	4	3	4	1	0.447214
5	5	3	3	3	4	1	0.500000
6	5	3	4	3	3	1	0.500000
7	5	3	4	3	5	1	0.500000
8	5	3	5	3	4	1	0.500000

Akurasi 0.9968051118210862

```
Confusion matrix:
[[233  0]
 [ 1 79]]
```

Dalam penerapan model KNN yang telah dilakukan, dapat terlihat bahwa tingkat akurasi mencapai nilai yang tinggi, yaitu sebesar 0.9968%. Hal ini menunjukkan bahwa model yang dibangun memiliki kemampuan yang sangat baik dalam memprediksi dengan benar. Artinya model KNN dapat dengan baik dalam mengklasifikasikan dan memprediksi konsumen yang layak dan tidak layak untuk menerima kredit.

Implementasi Menggunakan Python Flask

Flask adalah sebuah microframework aplikasi web berbasis Python yang ringan dan mudah digunakan. Framework ini memungkinkan pengembang untuk membangun aplikasi web dengan cepat dan efisien dengan menggunakan sintaks Python yang sederhana dan jelas.

Flask dikategorikan sebagai microframework karena memiliki inti yang kecil dan sederhana. Hal ini membuatnya mudah dipelajari dan dipahami, serta cocok untuk proyek-proyek kecil hingga menengah.

Kesederhanaan, fleksibilitas, dan kemudahan penggunaan menjadikan Flask framework yang ideal untuk berbagai jenis aplikasi web, mulai dari situs web sederhana hingga aplikasi web yang kompleks dengan banyak fitur.

Tampilan Home Page

Kelompok 3

Home

Input

SPK

Sistem Pengambilan Keputusan Kelayakan Penerima Kredit

Sistem untuk menentukan kelayakan
sebagai penerima kredit dengan
menggunakan metode klasifikasi

K-Nearest Neighbors

Pilih Pekerjaan

Pilih Jenis Pekerjaan



Tampilan Input dan Form Submit

Tanggungan

Jumlah Tanggungan

Penghasilan

Pilih Penghasilan Perbulan

Status Rumah

Pilih Status Rumah

Kelengkapan Berkas

Pilih Kelengkapan Berkas/Dokumen

Submit

Menampilkan Prediksi

Kelompok 3

Home

Input

SPK

Sistem Pengambilan Keputusan Kelayakan Penerima Kredit

Sistem untuk menentukan kelayakan
sebagai penerima kredit dengan
menggunakan metode klasifikasi

K-Nearest Neighbors



Hasil Prediksi

Pekerjaan: PNS/BUMN

Tanggungan: 3-4 Orang

Penghasilan: 2,5 s/d 3,5 juta rupiah

Status Rumah: Milik Instansi

Kelengkapan Berkas: Baik

Berdasarkan perhitungan di atas, sistem menyatakan pemohon **Memenuhi** mengajukan kredit.

Persentase probabilitas: **88.88888888888889%** (Diterima)

Referensi Kelompok 3 :

<https://drive.google.com/file/d/18JEurdO8PHurLNnmp-rVaJtfwyTo9dF7/view?usp=sharing>

<https://drive.google.com/file/d/1TeuwljHMUtQp5mHauh0Mrxn2nQq2EZYX/view?usp=sharing>

https://drive.google.com/file/d/10fB_S0YOeb3o_eygcYmDKk14KKt94Jn4/view?usp=sharing

<https://drive.google.com/file/d/1wwfcfyuj2OfhU6uLcbAabppsHP7ptA5C/view?usp=sharing>

https://drive.google.com/file/d/1t0EGj-fYm2eAXGlo_UXbT6VE7PxWpkWR/view?usp=sharing

<https://drive.google.com/file/d/1RJ6cKuu3vYE5OrZGx4I9pUAj9FW8qAq6/view?usp=sharing>

<https://drive.google.com/file/d/1BsPRHoyKkyvJ9ZAFw6MyWQIEYuvqG5x4/view?usp=sharing>

<https://drive.google.com/file/d/1SrcLL2xZetb3zc14LRKJB6a71leZbJKX/view?usp=sharing>

<https://drive.google.com/file/d/1GD3AvNCca02JTWx5OtOtOAI-zEm9BenS/view?usp=sharing>