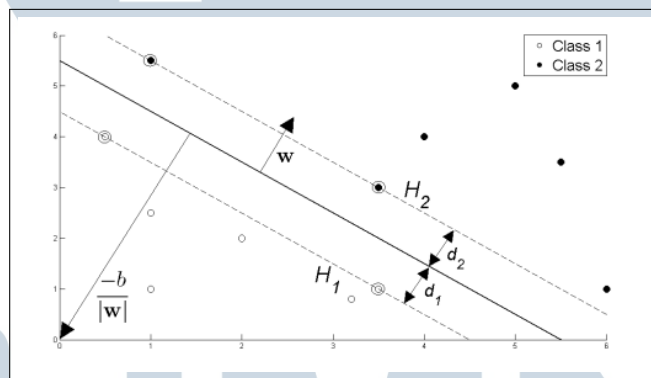


BAB 2 LANDASAN TEORI

2.1 Support Vector Machine

Merupakan salah satu algoritma pembelajaran mesin terbaik yang diusulkan pada tahun 1990-an *Support Vector Machine* ataupun *SVM* sebagian besar digunakan untuk pengenalan pola. *SVM* adalah salah satu algoritma *Supervised Machine Learning*, yang berkerja dengan diberikan *training data* sebagai contoh, setiap data tersebut ditandai sebagai salah satu dari banyak kategori, lalu setelah melewati proses *training SVM* dapat memprediksi kategori dari contoh yang baru. *SVM* pada dasarnya membangun *hyperplane* yang digunakan untuk mengklasifikasikan pola yang dapat dipisahkan secara linier [8].



Gambar 2.1. Hyperplane dan Support Vector Machine

Berdasarkan Gambar 2.1. *hyperplane* dapat dijelaskan dengan persamaan sebagai berikut [9].

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (2.1)$$

Margin yang digambarkan garis H_1 dapat dijelaskan sebagai berikut.

$$\mathbf{w} \cdot \mathbf{x}_i + b = +1 \quad (2.2)$$

Margin yang digambarkan garis H_2 dapat dijelaskan sebagai berikut.

$$\mathbf{w} \cdot \mathbf{x}_i + b = -1 \quad (2.3)$$

\mathbf{w} adalah normal untuk *hyperplane*, \mathbf{x} adalah *input vector* dan b adalah konstanta bias [9].

Algoritma SVM memakai sekelompok fungsi matematika yang diketahui selaku *kernel*. *kernel* memerlukan *data* selaku *input* lalu mengubahnya menjadi bentuk yang diinginkan. Memilih *Kernel Function* yang berbeda adalah hal yang penting dalam aspek klasifikasi berbasis *SVM*, *Kernel Function* yang umum digunakan termasuk *LINEAR*, *POLY*, *RBF*, dan *SIGMOID* [7].

- *Linear Kernel*
 $\langle x, x' \rangle$
- *Polynomial Kernel*
 $(\gamma \langle x, x' \rangle + r)^d$
- *Gaussian Radial Basis Function* atau *RBF*
 $\exp\left(-\gamma \|x - x'\|^2\right)$
- *Sigmoid Kernel*
 $\tanh(\gamma \langle x, x' \rangle + r)$

di mana [7]:

- x adalah vector input
- d adalah parameter *degree*
- r adalah *coef 0*
- γ adalah parameter *gamma* yang nilainya harus lebih dari 0.

Parameter penting lainnya dalam *SVM* adalah nilai C . nilai C disebut juga *Complexity Parameter* dan merupakan jumlah dari jarak semua titik-titik yang berada di sisi yang salah dari *hyper plane*. Pada dasarnya, *Complexity Parameters* adalah jumlah kesalahan yang dapat diabaikan selama proses klasifikasi. [8]

Parameter *SVM* lainnya adalah *Gamma*, Parameter *Gamma* yang tidak intuitif mendefinisikan lebar atau kemiringan *Kernel Function*. 'Kurva' dari batas keputusan menjadi sangat rendah ketika nilai *Gamma* rendah membuat wilayah keputusan sangat luas. 'Kurva' dari batas keputusan menjadi tinggi ketika nilai *Gamma* tinggi [10].

Sedangkan parameter *coef 0* adalah parameter yang menunjukkan seberapa jauh model tersebut dipengaruhi oleh fitur-fitur polinomial di tingkat yang lebih tinggi (*degree*). parameter ini hanya bekerja saat menggunakan *Polynomial Kernel*

dan juga *RBF. Degree* atau nilai d yang di maksud adalah tingkatan polinomial, yang mana hanya bekerja saat menggunakan *Polynomial kernel* [11].

Algoritma yang digunakan untuk menerapkan *Hyper Parameter Optimization* adalah *Grid Search*, *Grid Search* adalah algoritma *Brute Force* atau algoritma kasar yang mencari kombinasi terbaik berdasarkan subset yang sudah ditentukan dari *Hyper Parameter Space*[12]. *Grid Search* dipilih karena *Hyper Parameter Space* yang dimaksud dalam permasalahan ini masih terhitung kecil, juga pada penelitian yang dilakukan oleh Iwan Syarif, Adam Prugel-Bennett, dan Gary Wills menemukan bahwa *Grid Search* senantiasa menciptakan parameter yang mendekati optimal dengan kombinasi dari rentang yang diberikan.

Rentang nilai C dan γ yang digunakan adalah 10^{-3} sampai 10^3 dinilai cukup. Jika nilai γ terlalu besar maka area pengaruh *Support Vector* tersebut hanya mencakup *Support Vector* itu sendiri dan nilai C tidak dapat mencegah *Overfitting*, Jika nilai γ terlalu rendah maka model akan terlalu dibatasi dan tidak dapat menangkap kerumitan bentuk data. Meningkatkan nilai C tidak akan membantu karena tidak ada lagi nilai kesalahan yang dapat diabaikan atau tidak ada solusi lebih baik yang dapat ditemukan. Di samping itu nilai C yang terlalu kecil akan meningkatkan waktu prediksi [7].

2.2 Sistem Rekomendasi

Sistem rekomendasi adalah sistem yang dibuat bertujuan untuk memprediksi minat *user* pada suatu produk atau layanan yang tersedia dengan mengandalkan karakteristik dan nilai dari produk atau layanan tersebut. Faktor manusia yang ada dalam kepribadian, gaya hidup, juga pilihan *user* digunakan dalam proses personalisasi untuk membantu sistem mengenali produk atau layanan yang disukai *user* [13].

Beberapa contoh sistem rekomendasi yang sudah diterapkan di *E-Commerce* menurut J. Ben Schafer, Joseph Konstan, dan John Riedl adalah seperti di bawah [14].

Tabel 2.1. Contoh sistem rekomendasi pada E-Commerce

Contoh sistem rekomendasi pada <i>E-Commerce</i>			
Bisnis / Aplikasi	Antarmuka	Teknologi	Cara mengakses rekomendasi
Amazon			
textbfCustomer who Bought	Barang Serupa	Korelasi antar item, Data Pembelian	Navigasi Organik
Eyes	Surel	Berbasis Atribut	Kata kunci
Amazon.com Delivers	Surel	Berbasis Atribut	Pilihan
Book Matcher	Top N List	Korelasi antar orang, Likert	Daftar Permintaan
Customer Comments	Rata-rata penilaian, Teks komentar	Peringkat Agregat, Likert, Teks	Navigasi Organik
CDNOW			
Album Advisor	Barang Serupa, Top N List	Korelasi Antar item, Data Pembelian	Navigasi organik, Kata kunci
My CDMO	Top N List	Korelasi antar Orang, Likert	Navigasi organik, Daftar Permintaan
eBay			
Feedback Profile	Rata-rata penilaian, Teks komentar	Peringkat Agregat, Likert, Teks	Navigasi Organik

Beberapa contoh antarmuka yang digunakan pada situs *E-commerce*:

Browsing atau jelajah, seperti pada perdagangan tradisional seorang pelanggan bisa saja masuk ke sebuah toko dan meminta petugas untuk menyarankan sebuah *item*. Petugas akan menyarankan beberapa *item* kepada pelanggan dan pelanggan dapat menelusuri beberapa *item* yang sudah disarankan oleh petugas. Kelemahannya pada saran yang diberikan oleh petugas adalah ketergantungan kualitas saran yang diberikan oleh pengetahuan sang petugas [14].

Similar Item atau barang serupa, adalah modifikasi dari perdagangan tradisional, situs *E-commerce* dapat membuat rekomendasi yang lebih personal dan

khusus. *Item* yang ditampilkan dipilih berdasarkan *item* yang dimana pelanggan telah menunjukkan minatnya [14].

Email atau surel, rekomendasi juga bisa langsung dikirimkan melalui surel, contoh jika suatu *item* menjadi tersedia pada saat tertentu, maka sistem dapat mengirimkan surel kepada pelanggan sebagai pemberitahuan [14].

Text Comments atau teks komentar, adalah bagaimana situs *E-commerce* menyediakan rekomendasi berdasarkan teks komentar yang telah dibuat oleh pelanggan lain [14].

Average Rating atau nilai rata-rata, mirip seperti *Text Comments*, situs *E-commerce* menyediakan rekomendasi berdasarkan opini pelanggan lain yang berbentuk peringkat numerik [14].

Top-N List situs *E-commerce* mempelajari suka atau tidak suka dan juga ketertarikan pelanggan terhadap suatu *item*, lalu memberikan pelanggan daftar personal *item* yang mungkin memancing ketertarikan pelanggan [14].

Ordered Search Results ketika *Top-N List* memberikan rekomendasi yang terbatas pada jumlah yang ditentukan, *Ordered Search Results* memungkinkan pelanggan untuk terus melihat *item* yang sangat mungkin menarik bagi mereka [14].

Beberapa contoh teknik atau teknologi yang digunakan untuk menyediakan rekomendasi:

Non-Personalized Recommendations atau sistem rekomendasi yang dipersonalisasi memberikan rekomendasi produk berdasarkan apa yang dikatakan pelanggan-pelanggan lainnya. Rekomendasi ini tidak terbentuk berdasarkan satu pelanggan, maka setiap pelanggan mendapatkan rekomendasi yang sama [14].

Attribute-Based Recommendations menyarankan *item* atau produk berdasarkan atribut, sifat ataupun ciri dari produk tersebut, contohnya jika seorang mencari film dengan genre *Action*, maka sistem akan menyediakan film-film dengan genre *Action* [14]. *Attribute-Based Recommendations* biasanya manual karena pelanggan harus memasukkan atribut tersebut sendiri [14].

Item-to-Item Correlation atau korelasi *item*, rekomendasi yang dibentuk dari sedikit kumpulan *item* yang pelanggan telah menyatakan minatnya, contoh jika seorang pelanggan telah menyimpan beberapa produk ke keranjangnya, sistem mungkin saja menyarankan produk pelengkap dari produk yang sudah disimpan pelanggan.

People-to-People Correlation atau korelasi individu, memberikan rekomendasi berdasarkan korelasi seorang pelanggan dengan pelanggan lainnya, teknik ini juga sering di sebut sebagai *Collaborative Filtering* [14].

User Inputs atau masukan dari *user*, semua teknik rekomendasi di atas pastinya perlu beberapa bentuk *Input* yang bisa berasal dari *user* ataupun penyelenggara bisnis. bentuk-bentuk *Input* yang dimaksud adalah sebagai berikut:

- *Purchase data* adalah data apa saja produk yang sudah dibeli pelanggan.
- *Likert* adalah opini seseorang terhadap suatu produk dalam bentuk skala numerik.
- *Text* atau komentar tertulis yang ditujukan kepada pelanggan untuk dibaca.
- *Editor's choice* seleksi kategori produk yang dipilih oleh seorang editor manusia yang dipekerjakan oleh sang penyelenggarabisin atau situs *E-commerce* tersebut [14].

Beberapa cara untuk mengakses rekomendasi yang sudah disediakan:

Organic Navigation membutuhkan usaha paling sedikit untuk mengakses saran atau rekomendasi, rekomendasi menjadi bagian dari halaman informasi yang ada pada *Organic Navigation* atau navigasi organik [14].

Request Recommendation List tidak memerlukan banyak usaha pada *Request Recommendation List*, *user* hanya perlu untuk mengajukan permintaan atau *request* untuk mengakses rekomendasi yang sudah disediakan. *Selection Options* pada metode ini dibutuhkan lebih banyak interaksi dari *user*, biasanya *user* dibutuhkan untuk memilih dari serangkaian kriteria atau opsi yang sudah ditentukan [14].

Keyword/Freeform membutuhkan interaksi paling banyak dari *user*, *Keyword/Freeform* membutuhkan kata kunci yang disediakan oleh *user* untuk menghasilkan atau mengakses rekomendasi yang ada [14].

2.3 Akurasi dan F1-Score

Akurasi dan *F1-Score* adalah metrik yang paling populer untuk mengevaluasi klasifikasi biner dan *Confusion Metrics*. Akurasi mewakili rasio antara contoh yang diprediksi dengan benar dan semua contoh dalam kumpulan data [15]:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

F1-Score adalah metrik yang paling sering digunakan diantara keluarga parametrik *F-measures*. *F1-Score* didefinisikan sebagai rata-rata dari *Precision* and *Recall* dan memiliki fungsi sebagai berikut [16]:

$$\text{precision} = \frac{TP}{TP + FP} \quad (2.5)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (2.6)$$

$$F1 - \text{Score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (2.7)$$

singkatan [16]:

- TP = *True Positives*
- TN = *True Negatives*
- FP = *False Positives*
- FN = *False Negatives*

Dimana *TP* atau *True Positive* merupakan jumlah dari sampel positif yang diprediksi dengan benar, *TN* atau *True Negative* merupakan jumlah dari sampel negatif yang diprediksi dengan benar, *FP* atau *False Positive* merupakan jumlah dari sampel negatif yang salah diprediksi, serta *FN* atau *False Negative* merupakan jumlah dari sampel positif yang salah diprediksi [16].