BAB 2
LANDASAN TEORI

2.1 Machine Learning

Machine Learning merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) dan merupakan ilmu komputer yang berfokus pada penggunaan data serta algoritma untuk meniru cara manusia belajar dan melalui iterasi dapat meningkatkan akurasinya. *Machine learning* adalah program komputer untuk mengoptimalkan performa menggunakan data contoh atau pengalaman dengan mencari pola yang terdapat dalam data. Machine Learning terletak di antara *computer science* dengan statistik. *Machine learning* belajar bekerja dengan cara menemukan beberapa hubungan antara fitur dan variabel target, untuk menguji suatu model *machine learning* biasa diperlukan set pelatihan data (*training set*) dan test set yang merupakan set terpisah dari set pelatihan data. Algoritma diuji dengan melatih model menggunakan *training set*, dan diuji dengan *test set* agar model memberikan prediksi pada test set yang diuji akurasinya menggunakan variabel target yang terdapat dalam data.[6]

2.1.1 Klasifikasi

Klasifikasi adalah proses menemukan atau menemukan model (fungsi) yang membantu dalam memisahkan data dalam bentuk beberapa kelas kategori. Dalam klasifikasi, keanggotaan grup dari masalah diidentifikasi, yang berarti data dikategorikan dalam label yang berbeda sesuai dengan beberapa parameter dan kemudian label di prediksi untuk data tersebut.

Model turunan dapat didemonstrasikan dalam bentuk aturan "IF-THEN", *Decision Tree* atau jaringan saraf, dll. *Decision Tree* pada dasarnya adalah bagan alur menyerupai struktur pohon yang setiap simpul internal menggambarkan tes pada suatu atribut, dan cabang-cabangnya menunjukkan hasil tes. Pada proses klasifikasi data dapat dibagi dua label *discrete* atau lebih.

Dalam klasifikasi terdapat istilah *knowledge representation* yang merupakan sebuah metode yang dapat merepresentasikan sebuah dalam sebuah bentuk yang dapat dimengerti oleh sistem *machine learning*, biasa dapat terlihat dalam bentuk sekumpulan aturan (*rules*) atau sebuah training set. Klasifikasi dapat dibagi dua, yaitu klasifikasi biner dan klasifikasi *multiclass*. Dalam klasifikasi biner hanya

terdapat dua *class* sehingga klasifikasi dapat lebih mudah dipahami. Sedangkan pada klasifikasi *multiclass*, melibatkan penugasan objek pada lebih dari dua class. Dalam klasifikasi tugas utama yang dilakukan adalah memprediksi apakah suatu data masuk ke dalam kelas tertentu. Klasifikasi masuk ke dalam kategori *Supervised Learning* karena adanya target prediksi yang diberikan kepada algoritma, berbeda dengan *Unsupervised Learning* yang tidak memiliki label atau target yang diberikan sebagai data. *Unsupervised learning* hanya perlu memasukkan data karena *unsupervised learning* dengan tujuan mencari kesamaan dari data yang di-input menggunakan metode *clustering*. *Clustering* adalah metode mengumpulkan data berdasarkan kesamaan data tersebut. [7]

2.2 Random Forest

Random Forest merupakan salah satu algoritma yang digunakan dalam mengklasifikasi dan regresi. Random Forest masuk dalam kategori *Supervised Learning* dan *Ensemble*. *Supervised* karena dibutuhkannya target variable untuk pembuatan model dan *Ensemble* karena *Random Forest* merupakan kumpulan dari *Decision Trees* yang disebut *Estimators* dan setiap *Decision Tree* tersebut memiliki prediksinya masing-masing. Model Random Forest menggabungkan hasil prediksi tersebut untuk menghasilkan hasil prediksi yang lebih akurat.

Tujuan dari *Random Forest* adalah untuk memperbaiki kekurangan yang ada di dalam *Decision Tree*, kekurangan tersebut adalah kasus *overfitting* pada saat training. Dengan *Random Forest*, kemungkinan dari data yang tidak dapat dieksplorasi oleh *Decision Tree* dapat tercapai oleh *Random Forest*, alasan lain untuk memilih *Random Forest* adalah kemampuannya untuk mengolah data yang berskala besar. [8]

2.2.1 Metode-metode Random Forest

Dalam Random Forest terdapat beberapa metode yang membangun model tersebut yakni:

1. Bagging

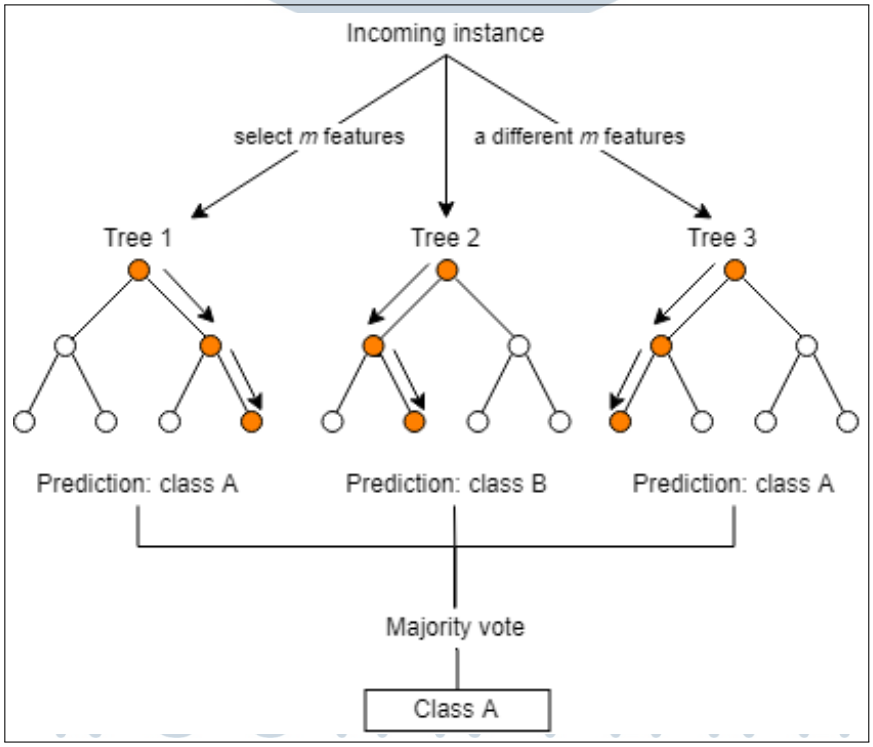
   Sebuah model memiliki sebuah dataset yang diproses ke dalam model dengan membentuk training data dan test data. Training data kemudian dibagi untuk membentuk subset training data yang lebih kecil untuk setiap *Decision Tree*, setiap subset memiliki kombinasi yang unik dan berbeda. Proses

membuat subsampling tersebut disebut dengan *Bootstrap Aggregating* dan biasa disingkat dengan nama *Bagging*.

2. Training Estimator
   Proses ini merupakan proses pembuatan beberapa *Decision Trees* dan dilatih dengan subset data fitur dan training untuk pembuatan model *Estimators*. Dalam proses ini *Decision Trees* atau *Estimators* tidak mengalami proses pruning yang biasa dialami oleh *Decision Tree* biasa.

3. Inference dengan agregasi prediksi estimators
   Untuk mendapat prediksi pada contoh data, maka diperlukannya fitur-fitur yang bersifat relevan pada setiap *Estimators*. Setelah mendapatkan prediksi dari setiap *Estimators* yang lalu digabung untuk menghasilkan prediksi secara keseluruhan. Dalam kasus metode klasifikasi, vote mayoritas digunakan untuk menentukan hasil prediksi, dan jika dalam kasus regresi maka digunakannya nilai rata-rata dari prediksi yang telah dibuat *Estimators*.

Berikut adalah sebuah graph yang menunjukan metode tersebut.



Gambar 2.1. Graph Metode RF

Sumber:Thomas Wood, 2010 [9]

2.2.2 Klasifikasi dalam Random Forest

Terdapat beberapa metode dalam pembuatan *Estimator* di *Random Forest*. Metode tersebut mendefinisikan cara pembuatan *trees* berdasarkan keputusan algoritma dalam memilih cabang pohon. Terdapat 2 cara yaitu:

1. Gini Impurity

   Gini impurity mengukur frekuensi sebuah elemen dalam dataset yang masuk ke dalam kategori label yang salah ketika sedang dipilih secara acak. Seperti namanya, metode ini mengukur kemurnian fitur, nilai minimum dari *Gini Index* adalah 0. Ini dapat disebut ketika sebuah node disebut murni, yang berarti bahwa elemen yang berada di dalam node tersebut adalah class yang bersifat unik dan dengan demikian node tersebut berhenti memecahkan diri. Untuk mencari pembelahan yang paling optimum, maka dicari nilai *Gini Index* yang paling rendah. Berikut adalah formula dari *Gini Index*:

   $$Gini = 1 - \Sigma_{i=1}^{C}(p_i)^2 \qquad \text{Gini Impurity Formula} \qquad (2.1)$$

2. Entropy

   Metode *Entropy* adalah sebuah metode yang mengukur informasi yang mengindikasi kekacauan antara fitur dengan target. Mirip dengan *Gini Index*, pemecahan yang optimum tercapai dengan memilih fitur dengan nilai *Entropy* yang paling rendah. Nilai maksimal dari *Entropy* dapat dicapai jika probabilitas dua kelas sama dan node termasuk pure jika nilai *entropy* berada dalam nilai minimum yaitu 0. Berikut adalah formula dari *Entropy*:

   $$E(S) = 1 - \Sigma_{i=1}^{C} - p_i \log p_i \qquad \text{Entropy Formula} \qquad (2.2)$$

2.3 Swarm Intelligence

Swarm Intelligence adalah salah satu teknik kecerdasan buatan yang berlandaskan kepada perilaku kolektif (*collective behaviour*) pada sistem yang terdesentralisasi dan dapat mengatur dirinya sendiri (*self-organizing*). Sistem yang memanfaatkan *Swarm Intelligence* biasanya merupakan sebuah populasi yang terdiri atas anggota berupa agen yang sederhana, yang berinteraksi secara lokal dengan sesama anggota, dan juga berinteraksi dengan lingkungan. Walaupun pada umumnya tidak ada struktur kendali secara terpusat (*centralized*) yang mendikte

bagaimana masing-masing individu bertindak, namun interaksi secara lokal (di antara anggota) seringkali menuju pada pembentukan (*emergence*) perilaku global. [10]

2.3.1 Sifat Dasar Swarm Intelligence

Sifat-sifat dari *Swarm Intelligence* adalah sebagai berikut:

1. Terdiri dari banyak *individual agent*.
2. Individu yang relatif bersifat homogen (Baik yang memiliki sifat yang sama maupun yang memiliki sifat atau tipe yang berbeda).
3. Interaksi antar individu didasari dari aturan perilaku sederhana yang memanfaatkan informasi lokal baik secara langsung atau melalui lingkungannya.

Kemampuan setiap individu melakukan interaksi dengan sesama individu maupun dengan lingkungannya dalam *swarm intelligence* disebut dengan *self organizes*. Hal ini menunjukkan bahwa setiap individu dapat mengatur dirinya sendiri.

2.3.2 Prinsip Dasar Swarm Intelligence

Prinsip-prinsip dasar Swarm Intelligence adalah sebagai berikut:

1. Prinsip kedekatan: setiap individu harus dapat menyesuaikan ruang dan waktu dengan perhitungan sederhana.
2. Prinsip kualitas: setiap individu harus dapat menanggapi faktor lingkungan.
3. Prinsip respon beragam: setiap individu tidak seharusnya melakukan kegiatannya pada jalur yang terlalu sempit.
4. Prinsip stabilitas: populasi tidak harus mengubah modus perilakunya setiap kali perubahan lingkungan.
5. Prinsip adaptasi: populasi harus mampu mengubah modus perilaku ketika komputasi bernilai penting atau berharga.

2.4 Ant Colony Optimization

Ant Colony Optimization adalah sebuah algoritma yang terinspirasi oleh sistem kerja semut dalam mencari makanan. Semut menggunakan sebuah hormon yang disebut dengan *Pheromone*, setiap semut dapat memberikan *Pheromone* untuk meninggalkan jejak pada jalan yang ditempuh oleh semut yang menemukan makanan. Hormon ini digunakan untuk memberi arah kepada teman-temannya. Bau hormon berakumulasi jika teman-teman semut yang lain mengikuti jalur yang sama dan meninggalkan hormon sehingga semakin pekat. Keuntungan algoritma ini adalah semut mulai dengan menjelajahi banyak jalan dan mencari jalan yang optimal.[11]

Dengan menyimulasikan aturan-aturan tersebut kita dapat membuat simulasi semut yang dapat memilih jalan terbaik. Dalam penelitian ini semut membangun aturan dari jalur yang mereka jelajahi. Aturan yang dikembangkan oleh semut virtual menjadi aturan klasifikasi yang menentukan prediksi dari program.

Untuk membuat simulasi semut dengan algoritma ACO diperlukan beberapa aturan dan rumus perhitungan, pertama *Entropy Level*, *Probability*, *Pheromone (Evaporation)*. Ketiga hal tersebut adalah kunci dari algoritma ACO. Algoritma ini merupakan algoritma yang lumayan mudah diterapkan dan memiliki potensi akurasi yang tinggi.[12]

2.4.1 Pheromone

Pada algoritma *Ant Colony Optimization*, *Pheromone* merupakan hal yang berkaitan dan sangat penting dalam algoritma *Ant Colony Optimization* yaitu *Pheromone*. *Pheromone* adalah mekanisme utama dalam algoritma ini yang merepresentasikan ciri kerja semut. *Pheromone* sendiri dapat dibagi menjadi 2 formula dalam algoritma ini yaitu:

$$\tau_{i,j} \leftarrow \tau_{i,j} + \Delta\tau^{k} \qquad \text{Pheromone Update Formula} \qquad (2.3)$$

$$\tau_{i,j} \leftarrow (1-p)\tau_{i,j}, j; \forall (i,j) \in A \qquad \text{Pheromone Evaporation Formula} \qquad (2.4)$$

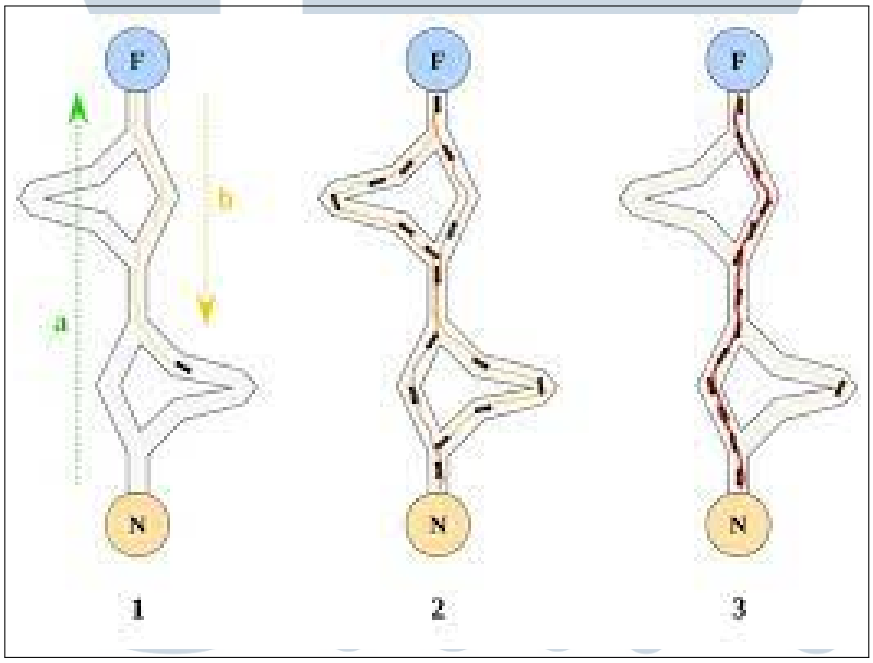
$\tau =$ Pheromone Level

p = Evaporation Rate

i, j = Path

k = Ant Index

Pheromone Update Formula adalah rumus yang digunakan untuk memperbarui nilai dari *Pheromone* sebuah jalan yang telah dilalui oleh semut. Lalu *Pheromone Evaporation Formula* adalah rumus yang digunakan untuk menyimulasikan evaporasi dari *Pheromone* saat tiap iterasi dalam algoritma *Ant Colony Optimization*.



Gambar 2.2. Graph Simulasi ACO Pheromone

Sumber:Fardad Farokhi, 2010 [13]

2.4.2 Probability

Pada algoritma ini setiap semut memiliki kebebasan untuk memilih jalan dengan tujuan memberikan semut kebebasan eksplorasi. Untuk setiap iterasi semut, cara pemilihan jalan bergantung kepada *Heuristic Function* dan juga *Pheromone*.

Probability di sini adalah metode yang digunakan untuk menghitung probabilitas pengambilan jalan berdasarkan kedua faktor tersebut dengan bias yang dapat diatur oleh penguji algoritma. Berikut adalah formula *Probability*:

$$P_{i,j} = \tau_{i,j}^{\alpha} \eta_{i,j}^{\beta} \quad \text{Probability Formula jika } j \in N_i^{(k)} \tag{2.5}$$

$$p_{i,j} = 0 \quad \text{Probability Formula jika } j \notin N_i^{(k)} \tag{2.6}$$

τ = Pheromone Level

η = Heuristic value

P = Probability of choosing paths

A = Attribute

α = Parameter to control the relative weight of the pheromone

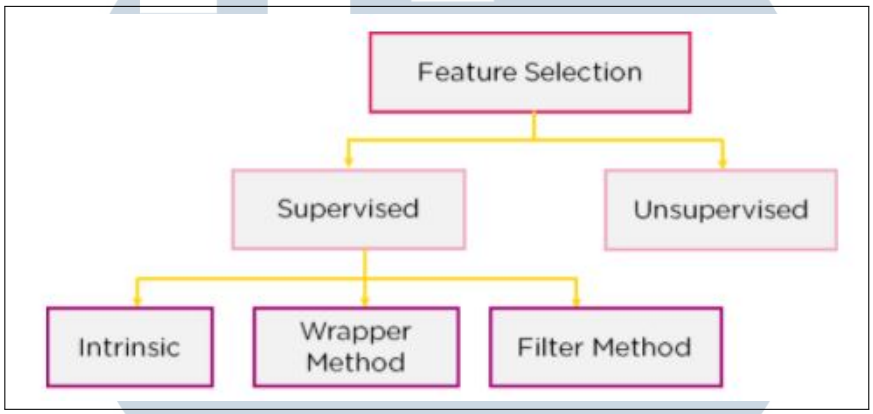
β = Parameter to control the relative weight of the heuristic

2.5 Feature Selection

Feature Selection adalah sebuah metode yang digunakan untuk mengurangi *variable input* ke dalam model dengan memilih data yang bersifat relevan serta menghilangkan *noise* dalam data. Metode merupakan proses otomatis pemilihan data fitur yang bersifat relevan untuk model machine learning berdasarkan masalah yang ingin diselesaikan.[14] *Feature Selection* dapat dibagi dua berdasarkan tipenya yaitu:

1. *Supervised Model*:
   Supervised feature selection adalah metode yang menggunakan output label class untuk feature selection. Metode ini menggunakan target variables untuk mengidentifikasi variable yang dapat menambah akurasi atau efisiensi model.

2. *Unsupervised Model*:
   Unsupervised feature selection adalah metode yang tidak menggunakan output label class untuk feature selection. Metode ini biasa digunakan untuk unlabelled data. Metode yang digunakan dalam *Unsupevised* model adalah *clustering* data untuk mencari kesamaan antar data.



Gambar 2.3. Graph pembagian Feature Selection

Sumber:Kartik Menon,2021 [15]

2.5.1 Teknik Feature Selection Supervised

Didalam metode Supervised model sendiri terdapat tiga metode yakni:

1. Metode Filter
   Dalam metode ini, fitur-fitur dihapus sesuai dengan relasi dan korelasi fitur terhadap output. Dengan menggunakan korelasi untuk memeriksa jika fitur tersebut memiliki korelasi yang positif atau negatif terhadap output labels dan memilih aksi sesuai dengan hasil yang didapat. Metode untuk menilai korelasi dapat dicapai dengan menggunakan salah satu dari metode seperti Eg. Information Gain, Chi-Square Test, Fisher's Score, etc.

2. Metode Wrapper
   Dalam metode *wrapper*, data dibagi menjadi subsets dan melatih (train) model menggunakan subset tersebut. Berdasarkan output dari model, metode ini menambah atau mengurangi fitur dan melatih model lagi. Dengan begini, metode ini membentuk sebuah subset menggunakan pendekatan *greedy* dan mengevaluasi akurasi dari kombinasi probabilitas dari fitur. eg: Forward Selection, Backwards Elimination, etc.

3. Metode Intrinsic

   Metode ini menggabungkan kualitas dari kedua metode Filter dan Wrapper untuk mencari subset fitur terbaik.