BAB 2
LANDASAN TEORI

2.1 Decision Tree

Decision tree method adalah alat statistik yang kuat untuk klasifikasi, prediksi, interpretasi, dan manipulasi data yang memiliki beberapa aplikasi potensial dalam penelitian.[9] *Decision tree models* biasanya digunakan untuk berbagai tugas, termasuk :

- *Variable selection*
  Seiring dengan semakin lazimnya penyimpanan data terkomputerisasi, jumlah variabel yang dilacak dalam pengaturan klinis telah meningkat secara signifikan. Banyak dari variabel-variabel ini hanya sesekali relevan, sehingga lebih baik untuk menghilangkannya dari aktivitas penggalian data. Metode *decision tree* dapat mengidentifikasi variabel input yang paling relevan untuk digunakan dalam model pohon keputusan, membantu perumusan hipotesis klinis dan mengarahkan penelitian selanjutnya. Pendekatan ini mirip dengan pemilihan variabel bertahap yang ditemukan dalam analisis regresi.

- *Assessing the relative importance of variables*
  Setelah mengidentifikasi sekumpulan variabel yang relevan, para peneliti sering kali berusaha untuk memahami peran penting yang dimainkan oleh variabel-variabel ini. Tingkat kepentingan variabel biasanya ditentukan dengan menilai seberapa besar akurasi model (atau kemurnian simpul dalam tree) berkurang ketika sebuah variabel dihilangkan. Secara umum, semakin besar dampak dari sebuah variabel terhadap jumlah record yang lebih besar, maka variabel tersebut dianggap semakin penting.

- *Handling of missing values*
  Menangani data yang hilang dengan mengecualikan kasus-kasus dengan nilai yang hilang adalah pendekatan yang umum dilakukan namun tidak tepat. Metode ini tidak efisien dan dapat menimbulkan bias dalam analisis. Analisis *decision tree* menawarkan dua pendekatan alternatif untuk menangani data yang hilang. Pertama, ia dapat memperlakukan nilai yang hilang sebagai kategori yang terpisah, yang memungkinkan mereka untuk dianalisis bersama kategori lainnya. Atau, *decision tree model* dapat dibangun dengan

menggunakan variabel dengan banyak nilai yang hilang sebagai variabel target untuk prediksi, dan mengganti nilai yang hilang tersebut dengan hasil prediksi.

- *Prediction*
  Salah satu aplikasi utama dari *decision tree model* adalah memprediksi hasil di masa depan. Dengan memanfaatkan data historis dan membangun *tree model*, para peneliti dapat dengan mudah meramalkan hasil untuk catatan yang akan datang.

- *Data manipulation*
  Dalam penelitian, adalah hal yang umum untuk menemukan terlalu banyak kategori untuk satu variabel kategorikal atau data kontinu yang sangat miring. Decision tree model dapat membantu dengan menyarankan cara menyederhanakan variabel kategorikal menjadi jumlah kategori yang lebih mudah dikelola atau cara membagi variabel yang condong ke dalam rentang variabel.

2.2 CART Algorithm

Classification and Regression Trees (CART) diperkenalkan oleh Breiman pada tahun 1984. Konstruksi *classification tree* didasarkan pada pemisahan biner dari atribut, menggunakan ukuran seperti indeks Gini. Tidak seperti algoritma berbasis Hunt lainnya, *CART* juga dapat digunakan untuk analisis regresi melalui pohon regresi. Analisis regresi memungkinkan peramalan variabel dependen berdasarkan variabel prediktor selama periode waktu tertentu. *CART* menggunakan berbagai kriteria pemisahan variabel tunggal dan multi-variabel untuk menentukan titik pemisahan terbaik. *CART* menyimpan data di setiap titik untuk mengoptimalkan proses pemisahan. *SALFORD SYSTEMS* mengimplementasikan versi *CART* yang telah disempurnakan, mengatasi keterbatasannya dan menghasilkan pengklasifikasi pohon keputusan modern dengan akurasi tinggi dalam klasifikasi dan prediksi. [10] CART menggunakan kriteria Gini impurity untuk mengukur kualitas pemisahan. Gini impurity mengukur probabilitas kesalahan klasifikasi jika suatu elemen dipilih secara acak.

$$\text{Gini}(S) = 1 - \sum_{i=1}^{|C|} p_i^2 \tag{2.1}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{2.2}$$

$$\Delta\text{Gini}(S,A) = \text{Gini}(S) - \left(\frac{|S_1|}{|S|}\text{Gini}(S_1) + \frac{|S_2|}{|S|}\text{Gini}(S_2)\right) \tag{2.3}$$

$$\Delta\text{Variance}(S,A) = \text{Variance}(S) - \left(\frac{|S_1|}{|S|}\text{Variance}(S_1) + \frac{|S_2|}{|S|}\text{Variance}(S_2)\right) \tag{2.4}$$

Untuk tugas regresi, CART menggunakan kriteria varians untuk mengukur homogenitas. Algoritma dimulai dengan memilih atribut dan nilai *split* yang meminimalkan *impurity* atau varians dari *subset* yang dihasilkan. Proses ini diulang secara rekursif untuk setiap *subset* hingga mencapai kedalaman maksimum pohon atau tidak ada lagi pengurangan *impurity* atau varians yang signifikan. Hasil akhirnya adalah pohon keputusan yang dapat digunakan untuk memprediksi label kelas atau nilai kontinu dari data baru.

2.3 ID3 Algorithm

Iterative Dichotomiser 3 (ID3) adalah *straightforward decision tree learning* algoritma yang diperkenalkan oleh Quinlan Ross pada tahun 1986. Algoritma ini mengikuti pendekatan *top-down*, pendekatan serakah berdasarkan algoritma Hunt. Konsep utama dari ID3 adalah membangun *decision tree* dengan menguji setiap atribut pada setiap simpul pohon. Untuk memilih atribut yang paling berguna untuk klasifikasi, ID3 menggunakan metrik perolehan informasi. Namun, metrik ini hanya bekerja dengan atribut kategorikal dan mungkin tidak memberikan hasil yang akurat dengan adanya noise. Pra-pemrosesan data sangat penting sebelum membangun *decision tree model* menggunakan ID3.[10]
Proses pemilihan atribut didasarkan pada pengukuran entropi dan gain informasi. Entropi mengukur ketidakpastian atau kekacauan dalam data, sedangkan gain informasi mengukur pengurangan entropi yang dihasilkan dari pemisahan data berdasarkan atribut tertentu. Algoritma ID3 dimulai dengan menghitung entropi dari seluruh dataset, kemudian untuk setiap atribut, menghitung *gain* informasi yang

dihasilkan dari pemisahan data berdasarkan atribut tersebut.

$$\text{Entropy}(S) = -\sum_{i=1}^{|C|} p_i \log_2 (p_i) \tag{2.5}$$

$$\text{Gain}(S,A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v) \tag{2.6}$$

Atribut dengan *gain* informasi tertinggi dipilih sebagai node keputusan, dan proses ini diulang secara rekursif untuk setiap subset data yang dihasilkan, hingga semua data dalam subset memiliki label yang sama atau tidak ada atribut yang tersisa untuk dipilih. Hasil akhirnya adalah pohon keputusan yang dapat digunakan untuk mengklasifikasikan data baru.