Aplikasi Pohon pada Model Machine Learning Prediksi Gender

Muhammad Zaydan Athallah - 13521104[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*author@itb.ac.id*

***Abstract*—Machine Learning adalah sesuatu yang semua orang pernah dengar di abad ini. Dari naiknya social media yang menggunakan machine learning seperti Tik-Tok hingga algoritma rekomendasi Youtube. Perkembangan teknologi juga diikuti satu konsekuensi yang tidak dapat dihindari yaitu berkembangnya kebutuhan data kita. Data dapat digunakan untuk berbagai hal, salah satu contoh aplikasinya yang paling terlihat adalah untuk digunakan dalam konteks Machine learning. Di makalah ini, akan membahas tentang salah satu bentuk dari supervised machine learning yaitu Decision Tree.**

***Keywords*—Decision Tree, Random Tree Forest, Pohon.**

I. INTRODUCTION

Tidak bisa dipungkiri, perkembangan teknologi komputasi akan diikuti dengan membesarnya data yang dihasilkan oleh manusia. Tahun lalu volume data total manusia adalah 33 Zettabyte. Angka tersebut diproyeksikan untuk naik hingga.

Dengan data yang semakin banyak, menjadikan industri terdorong untuk menggunakan data tersebut untuk membuat pilihan-pilihan yang didasarkan oleh data. Untuk memproses sebuah data, dibutuhkan model untuk menjelaskan data tersebut. Model dalam konteks data science adalah diagram deskriptif yang menjelaskan hubungan antara berbagai variabel di dalam sebuah dataset.

Machine learning adalah topik hangat di industri dan dunia riset. Kecepatan perkembangan dan kompleksitas dari bidang ini susah diikuti bahkan untuk para profesional di dalam bidang ini. Machine learning secara umum dapat dibagi kedalam dua kategori. Kategori pertama adalah *supervised learning* yang biasa dipakai jika yang inginkan adalah model yang dapat dijelaskan dengan bahasa manusia. Unsupervised learning digunakan untuk menciptakan model yang lebih kompleks dan lebih abstrak.

Salah satu model yang sederhana dan sering digunakan di data science adalah model *Random Tree Forest*. Model ini senang dipakai oleh para *data scientist* karena modelnya mudah dijelaskan dan performanya yang relatif cukup bagus untuk data diskrit.

Meskipun model *Random Tree Forest* paling efektif digunakan untuk dataset yang tipe datanya diskrit, model ini juga dapat digunakan untuk tipe data kontinu.

Makalah ini akan membahas implementasi teori graf dalam model *Decision Tree* dan *Random Tree Forest* dan menjelaskan aplikasinya dalam dunia nyata.

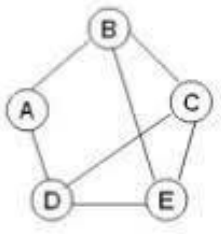
II. TEORI DASAR

A. Graf (Graph)

Graf G didefinisikan sebagai pasangan himpunan (V,E) ditulis dengan notasi G = (V,E) yang dalam hal ini V adalah himpunan tidak kosong dari simpul-simpul (vertices atau node) dan E adalah himpunan sisi (edges atau arcs yang menghubungkan) sepasang simpul Sesuai dengan definsi tersebut, maka sebuah graf dimungkinkan tidak mempunyai sisi, tetapi simpulnya tetap harus ada. Graf yang hanya mempunyai satu buah tanpa sebuah sisi pun dinamakan sebagai graf trivial. Selain dari graf trivial, juga terdapat berbagai jenis graf berdasarkan kategori-kategori. Jenis-jenis graf tersebut antara lain:

- Graf sederhana dan graf non-sederhana yang dikategorikan berdasarkan ada tidaknya loop atau gelang pada graf.
- Graf berarah dan graf tak berarah yang dikategorikan berdasarkan ada tidaknya orientasi arah pada graf.
- Graf khusus atau pohon yang memiliki sifat khusus

Dalam Pemrograman, Sebuah Graf dapat direpresentasikan dalam berbagai bentuk, seperti bentuk Adjacency List, Adjacency Matrix, Incidency Matrix, dan bentuk lainnya. Contoh Representasi Graf dalam bentuk Adjacency Matrix beserta bentuk dari graf seperti di bawah ini :



	A 0	B 1	C 2	D 3	E 4
A 0	0	1	0	1	0
B 1	1	0	1	0	1
C 2	0	1	0	1	1
D 3	1	0	1	0	1
E 4	0	1	1	1	0

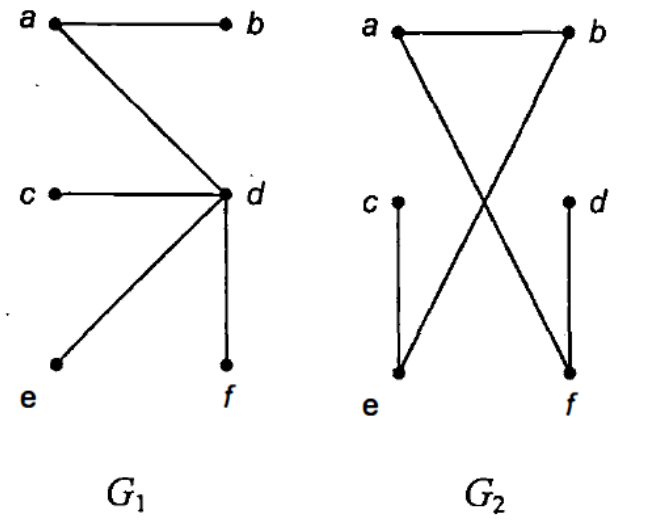
Gambar 1. Representasi Graf dalam Bentuk Adjacency Matrix.
Sumber : Google , diakses pada 1 Desember 2022.

B. Pohon (Tree)

Pohon merupakan sebuah graf yang memiliki beberapa aturan khusus, semua pohon adalah graf dan tidak semua graf adalah pohon.

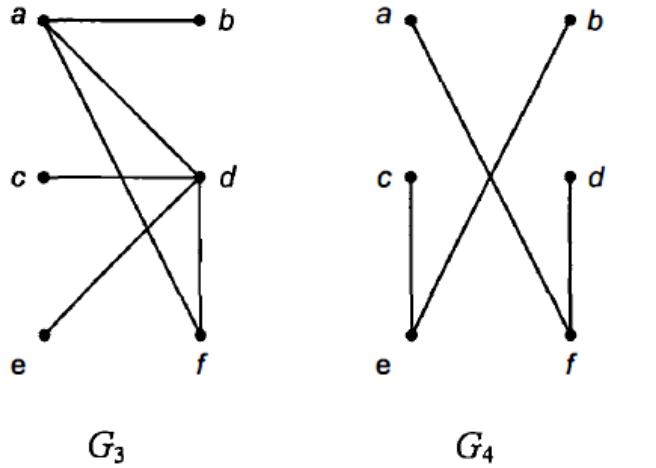
Pohon adalah sebuah graf tak berarah dan tidak memiliki sirkuit. Graf memiliki sirkuit apabila ada 1 simpul yang memiliki lintasan tidak terputus dimana lintasan dimulai melalui 1 sisi dan berakhir dengan masuk ke simpul melalui sisi yang berbeda daripada sisi awal.

Berikut 2 buah contoh gambar graf yang merupakan pohon dan 2 contoh graf yang bukan merupakan pohon sebagai perbandingan :



Gambar 2. Contoh pohon.

Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.



Gambar 3. Contoh Graf bukan pohon.

Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

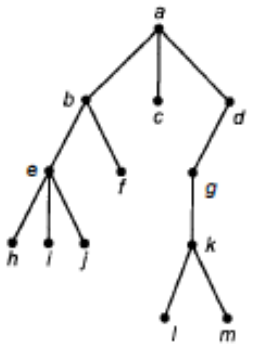
Kedua graf pada gambar 3 bukan merupakan pohon .Graf G_3 bukan merupakan pohon karena graf tersebut mengandung sirkuit a,d,f,a sedangkan G_4 bukan pohon karena graf tersebut tidak terhubung.

Misalkan G = (V,E) adalah graf tak-berarah sederhana dan jumlah simpulnya n. Maka, graf tersebut akan memiliki sifat-sifat (properti) sebagai berikut :

1. G adalah pohon.
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal.
3. G terhubung dan memiliki m = n-1 buah sisi.
4. G tidak mengandung sirkuit dan memiliki m = n -1 buah sisi.
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit.
6. G terhubung dan semua sisinya adalah jembatan.

C. Pohon Berakar

Pada kebanyakan aplikasi pohon, simpul tertentu diperlakukan sebagai akar (root). Keunikan pohon berakar dalam ilmu matematika diskrit adalah peletakkan akar pada pohon tersebut dimana akar diletakkan paling atas dan mengarah ke bawah, berbeda dengan di kehidupan nyata di mana akar sebuah tumbuhan akan bertumbuh dari bawah menuju ke atas. Ketika sebuah simpul ditetapkan sebagai akar, maka simpul – simpul lainnya dapat dicapai dari akar dengan memberi arah pada sisi-sisi pohon yang mengikutinya. Akar memiliki derajat-masuk sama dengan nol dan simpul-simpul lainnnya berderajat-masuk sama dengan satu. Simpul yang mempunyai derajat-keluar sama dengan nol disebut daun atau simpul terminal. Simpul yang mempunyai derajat-keluar tidak sama dengan nol disebut simpul dalam atau simpul cabang. Setiap simpul di pohon dapat dicapai dari akar dengan sebuah lintasan tunggal (unik). Berikut adalah contoh pohon berakar.



Gambar 4. Pohon Berakar

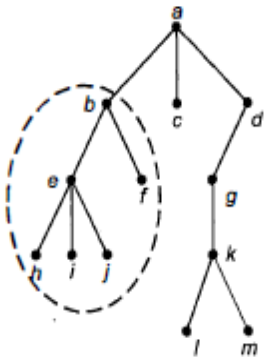
Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

Di bawah ini merupakan beberapa terminologi yang penting untuk pohon berakar. Untuk ilustrasi, pohon pada Gambar 4 dipakai sebagai contoh untuk menjelaskan terminologi yang dimaksudkan. Simpul – simpul pada pohon diberi label untuk mengacu simpul mana yang dimaksudkan.

1. Anak (child) dan Orang tua (parent)
   Misalkan a adalah sebuah simpul di dalam berakar. Simpul b dikatakan anak simpul a jika ada sisi dari simpul a ke b. Dalam hal demikian, a disebut sebagai orangtua (parent) y.
2. Lintasan (path)

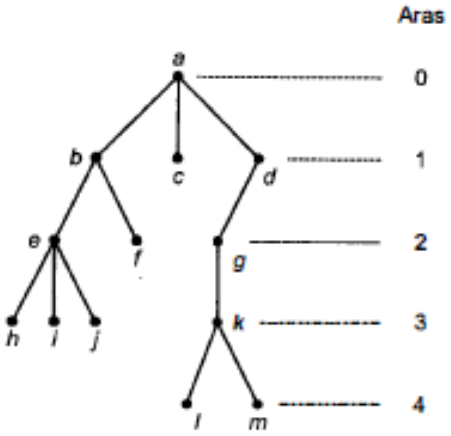
Lintasan dari simpul v1 ke simpul vk adalah runtunan simpul-simpul v1, v2,….. vk sedemikian sehingga vk adalah orang tua dari vk+1 untuk $1 \leq i \leq k$. Dari pohon pada Gambar 4 , lintasan dari a ke j adalah a, b, e, j. Panjang lintasan adalah jumlah sisi yang dilalui dalam suatu lintasan, yaitu k-1. Panjang lintasan dari a ke j adalah 3.

3. Keturunan (descendant) dan Leluhur (ancestor)
   Jika terdapat lintasan dari simpul x ke simpul y di dalam pohon, maka x adalah leluhur dari simpul y, dan y adalah keturunan dair simpul x. Pada Gambar 4 ,b adalah leluhur dari h , dan dengan demikian h adalah keturunan b.
4. Saudara kandung (sibling)
   Simpul yang memiliki orang tua sama disebut dengan saudara kandung satu sama lain. Pada Gmabar 4 , f adalah saudara kandng e, tetapi g bukan saudara kandung e, karena orang tua mereka berbeda.W
5. Upapohon (subtree)
   Misalkan x adalah sebuah simpul di dalam pohon T . Yang dimaksud dengan upapohon dengan x sebagai akarnya ialah upagraf T' = (V', E') adalah upapohon dari pohon pada Gambar 4 dengan V' {b,e,f,h,i,j} dan E' = { (b,e),(b,f),(e,h),(e,i),(e,j)} dan b adalah simpul akarnya Tedapat banyak upapohon di dalam pohon T. Dengan pengertian di atas, jika x adalah simpul maka akar tiap-tiap upapohon dari x disebut anak, dan x adalah orangtua setiap akar upapohon. Berikut ditampilkan contoh upapohon dalam pohon pada Gambar 4



Gambar 5. Pohon berakar dengan informasi aras
Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

6. Derajat (degree)
   Derajat sebuah simpul pada pohon berakar adalah jumlah upapohon (atau jumlah anak) pada simpul tersebut. Pada Gambar 4, derajat a adalah 3, derajat b adalah 2, derajat d adalah satu dan derajat c adalah 0. Maka derajat yang dimaksud di sini adalah derajat keluar. Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri. Pohon pada Gambar 4 berderajat 3, karena derajat tertinggi dari seluruh simpulnya adalah 3.W
7. Daun (leaf)
   Simpul yang berderajat nol (atau tidak mempunyai anak) disebut daun. Simpul h,i,j,f,c,l,m adalah daun.
8. Simpul Dalam (internal nodes)
   Simpul yang mempunyai anak disebut simpul dalam. Simpul d,e,g,dan k pada Gambar 4 adalah simpul dalam.
9. Aras (Level) atau Tingkat
   Akar mempunyai aras = 0, sedangkan aras simpul lainnya = 1 + panjang lintasan dari akar ke simpul tersebut. Beberapa literatur memulai nomor atas dari 0 , literartur lainnya dari 1. Sebagai konvensi, pada makalah ini penomoran aras dari 0. Berikut adalah gambar sebagai ilustrasi tentang aras suatu pohon berakar.

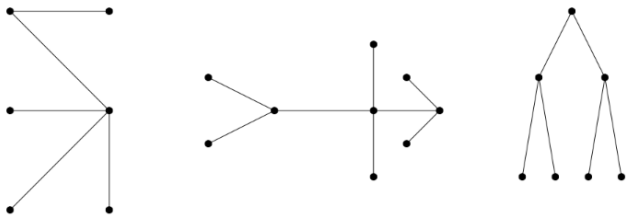


Gambar 6. Pohon berakar dengan informasi aras
Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

10. Tinggi (height) atau Kedalaman (depth)
    Aras maksimum dari suatu pohon disebut tinggi atau kedalam pohon tersebut. Dapat juga dikatakan tinggi pohon adalah panjang maksimum lintasan dari akar ke daun. Pohon pada Gambar 4 memiliki tinggi 4.

D. Hutan

Seperti pada arti umumnya, Hutan merupakan kumpulan dari lebih dari satu atau banyak pohon. Hal tersebut juga berlaku pada teorema graf ini. Dimana hutan merupakan kumpulan dari banyak bentukan pohon, seperti pada ilustrasi di bawah ini.



Gambar 7. Contoh Hutan dengan Tiga Pohon

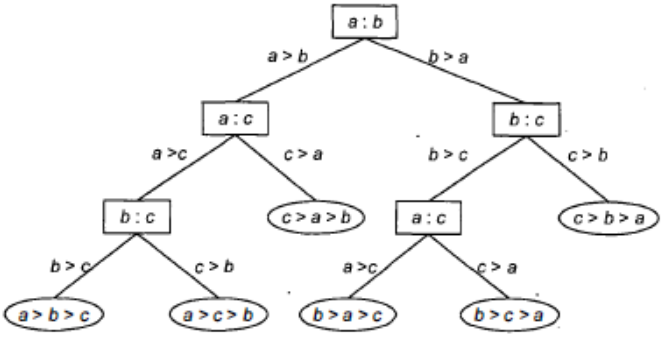
Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

F. Pohon m-ary

Pohon n-ary adalah pohon yang setiap simpul cabangnya mempunyai paling banyak n buah anak anak. Untuk m=2 maka pohon tersebut disebut pohon binary, untuk m=3 disebut trirary. Sebuah pohon m-ary dikatakan penuh jika semua simpul cabangnya mempunyai tepat m buah anak.

G. Pohon keputusan (decision tree)

Pohon keputusan adalah pohon berakar yang digunakan untuk memodelkan persoalan yang terdiri dari serangkaian keputusan yang mengarah ke solusi. Tiap simpul menyatakan suatu keputusuan, sedangkan tiap daun menyatakan solusi. Sebagai contoh kita inigin mengurutkan tiga buah bilangan a,b,dan c. Pohon keputusan persoalan ini dapat dilihat pada gambar di bawah ini.



Gambar 8. Pohon keputusan

Sumber : Munir, Rinaldi, Diktat Kuliah IF2120, Matematika Diskrit Edisi III, Program Studi Teknik Informatika, STEI, ITB, 2005.

Secara tidak langsung, kita telah memakai *decision tree* di kehidupan kita. Salah satu contoh kasus adalah dalam pemilihan makanan. Kasus lain adalah kategorisasi sebuah wilayah.

Decision Tree juga merupakan salah satu model dalam machine learning. Model *Decision Tree* adalah salah satu model paling sederhana dan paling mudah digunakan. Model ini membuat prediksi berdasarkan observasi sebuah objek dan membungkusnya kedalam sebuah *decision tree*.

Model *decision tree* dapat dibagi menjadi dua tipe.

1. Analisis *Classification tree* digunakan saat output yang diinginkan adalah sebuah data diskrit (contoh :yes/no)
2. Analisis *Regression tree* digunakan saat hasil yang diinginkan adalah sebuah angka riil. (contoh : harga properti)

Pada dasarnya kedua tipe ini mirip, perbedaannya hanya ada di prosedur dalam menentukan pemecahan nilai di simpul dalam.

III. RANDOM TREE FOREST

Pada Bab ini akan dibahas *Random Tree Forest* secara teknis, mulai dari pembuatan model *Decision Tree,* dan lainnya.

A. *Decision Tree*

Decision Tree adalah alat yang dipakai untuk klasifikasi dan prediksi. *Decision tree* adalah pohon n-ary dengan setiap daun merepresentasikan sebuah kelas atau nilai (sesuai dengan tipe *Decision tree* yang digunakan). Setiap simpul dalam merepresentasikan sebuah komparasi seperti (apakah umur > 15). Sebagai contoh, misalkan ada simpul yang melihat apakah Dia perempuan atau tidak, jika ya maka lintasan yang dilewati adalah anak di kanan simpul komparasi tersebut, jika tidak maka akan melewati anak di kiri simpul komparasi tersebut.

Pembuatan *decision tree* dimulai dengan membagi dataset yang merupakan simpul akar pohon menjadi dua subset yang merupakan anak dari simpul tersebut. Pembagian didasarkan pada seperangkat aturan pemisahan berdasarkan fitur-fitur yang terdapat di dalam dataset. Pembagian ini dilakukan terus menerus hingga tercipta pure subset dimana semua data didalam subset menghasilkan kelas/nilai tertentu. Proses induksi pohon keputusan inilah contoh dari *greedy algorithm* karena hanya memperhatikan hasil lokal yang paling optimal.

Sebagai contoh, misal kita punya dataset tinggi badan dan ukuran sepatu banyak untuk pria dan wanita :

sex	height	shoe
woman	160.00	40.0
woman	171.00	39.0
woman	174.00	39.0
woman	176.00	40.0
man	195.00	46.0
woman	157.00	37.0
woman	160.00	38.0
woman	178.00	39.0
woman	168.00	38.0
man	171.00	41.0
woman	165.00	39.0
man	175.00	44.0
woman	163.00	38.0
woman	158.00	37.0
…	…	…

Tabel 1. Dataset

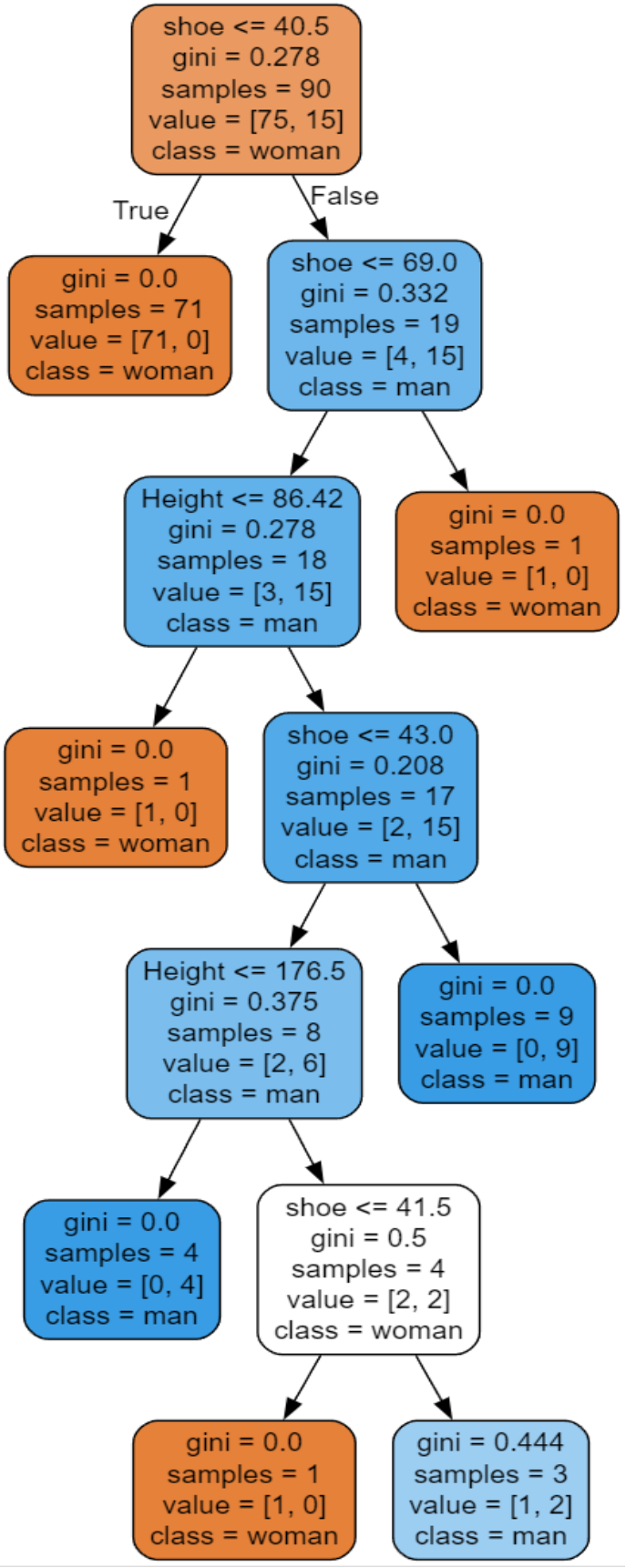
Sebelum membuat model machine learning kita akan melakukan *Data Preprocessing* agar data yang kita punya lebih bersih dan model yang dihasilkan menjadi lebih akurat.

sex	height	shoe
0	160.00	40.0
0	171.00	39.0
0	174.00	39.0
0	176.00	40.0
1	195.00	46.0
0	157.00	37.0
0	160.00	38.0
0	178.00	39.0
0	168.00	38.0
1	171.00	41.0
0	165.00	39.0
1	175.00	44.0
0	163.00	38.0
0	158.00	37.0
0	159.00	38.0
1	183.00	44.0
0	155.00	37.0
0	172.00	39.0
0	164.00	39.0
0	158.00	35.0
0	174.00	37.0
0	164.00	37.0
0	168.00	38.0
0	168.00	38.0
0	163.00	37.0
0	160.00	37.0
1	183.00	46.0
0	161.00	38.0
…	…	…

Tabel 2. Dataset Setelah Data Preprocessing

Data Preprocessing tersebut merubah man dan woman menjadi 0 dan 1 karena pada model tree membutuhkan hasil 0 dan 1. *Data Preprocessing* tersebut juga menghilangkan data yang memiliki nilai NaN dan juga data kosong. Setelah ini kita akan mendapatkan suatu model machine learning yang lebih akurat karena tidak ada data kotor saat dilakukan training.

Dari data tersebut dihasilkan model *Decision Tree* sebagai berikut :



Gambar 8. Model Machine Learning Decision Tree
Sumber : Dokumentasi Pribadi

Dari model di atas kita mendapatkan nilai *gini* yang memiliki nilai dari 0 sampai 0.5. *gini* adalah fungsi yang menentukan seberapa baik pohon keputusan dipecah.

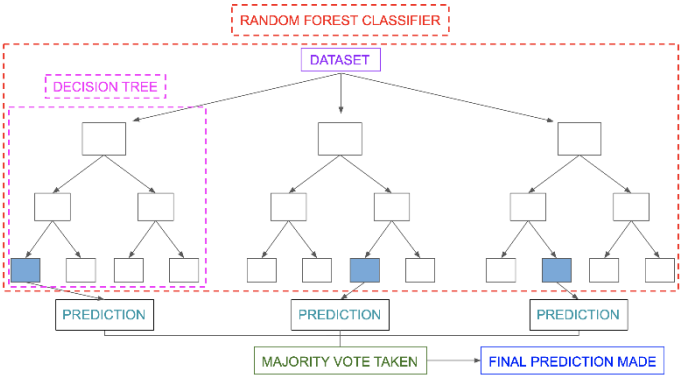
Dari pembagian data dan model tersebut didapatkan bahwa jika ukuran sepatunya kurang dari sama dengan 40.5 maka dia adalah wanita dan seterusnya.

Dengan *Decision Tree* tersebut, kita dapat memprediksi bahwa orang misterius X dengna ukuran sepatu 36 adalah Wanita.

Link Source Code :
https://github.com/zaydanA/MakalahMatematikaDiskrit.git

B.Random Tree Forest

Random Tree Forest adalah teknik pembuatan model yang didasari oleh *decision tree*. *Random Tree Forest* menggunakan hasil dari banyak *decision tree* untuk menghasilkan keluaran dengan merata-ratakan hasil dari semua *decision tree*.



Gambar 9. Simplified *Random Tree Forest*
Sumber : https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

Random Tree Forest pertama dibuat dengan memecah dataset menjadi sampel yang unik untuk setiap *decision tree* yang akan dibuat. Setelah itu, *estimator* (*decision tree* dalam konteks *random tree forest*) dibuat sesuai dengan sampel data yang unik untuk dirinya mereka sendiri. Sampel data tersebut dipilih secara random sehingga muncul kata *random* di dalam *random tree forest*. Setelah *estimator* selesai diciptakan, akan dilakukan *bootstrap aggregating*. *Bootstrap aggregating* atau yang biasa disebut *bagging/majority voting* adalah proses untuk mencari mean dari hasil prediksi masing estimator di dalam *random tree forest*. Secara matematis dapat ditulis sebagai berikut :

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Proses ini dapat menghasilkan model yang lebih baik karena mengurangi variansi dalam model tanpa mengurangi bias. *Random Tree Forest* secara keseluruhan tidak sensitif asalkan setiap *estimator* di dalam *random tree forest* tidak berkorelasi antara satu sama lain.

Ketidakpastian sebuah model dapat didapatkan dengan persamaan sebagai berikut:

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$

D. Aplikasi dalam dunia nyata

Beberapa model yang sudah disebutkan didalam makalah ini dapat diaplikasikan kedalam beberapa hal. Salah satu contoh riil adalah pengesahan credit core sebuah bank kepada nasabahnya. Selain dalam modeling, model *decision tree* dan *random tree forest* sudah diimplementasikan kedalam library python yang bernama scikit learn. Model-model sederhana ini dapat digunakan sebagai langkah awal mempelajari model-model machine learning yang lebih kompleks

IV. KESIMPULAN

Di dalam dunia ini, banyak hal disimplifikasi menjadi objek diskrit untuk mempermudah pemrosesan.Perwujudan model ini dapat menjadi bantuan besar untuk digunakan dalam hal yang bermanfaat bagi kehidupan sehari-hari.Tentunya, akan ada banyak kasus yang lebih kompleks dan memerlukan algoritma yang lebih mutakhir.

V. CONCLUSION

Saya, sebagai penulis ingin mengucapkan terima kasih kepada :

1. Bapak Dr. Ir. Rinaldi Munir, M. T., Ibu Dra. Harlili S., M. Sc., Ibu Fariska Zakhralativa Ruskanda, S.T., M.T., atas bimbingannya dalam mata kuliah IF2120 Matematika Diskirt terutama untuk Pak Munir yang telah mengajar di K-01
2. Keluarga dan teman-teman yang turut membantu dan mendukung saya dalam menjalani perkuliahan
3. Ilmuwan lampau yang telah menciptakan fondasi machine learning sekarang.

REFERENCES

[1] Munir,Rinaldi. 2022. Graf (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit. Merupakan slide bahan ajar perkuliahan. Diakses pada tanggal 3 Desember 2022.
[2] Munir,Rinaldi. 2022. Pohon (Bag. 1): Bahan Kuliah IF2120 Matematika Diskrit. Merupakan slide bahan ajar perkuliahan. Diakses pada tanggal 3 Desember 2022.
[3] Munir,Rinaldi. 2022. Pohon (Bag. 2): Bahan Kuliah IF2120 Matematika Diskrit. Merupakan slide bahan ajar perkuliahan. Diakses pada tanggal 3 Desember 2022.
[4] OSF | Dataset "Height and shoe size" , Diakses pada tanggal 8 Desember 2022, 21.10 WIB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2022

Muhammad Zaydan Athallahah/13521104