Analisis Keputusan Medis Melalui Penerapan Random Forest pada Data dengan Varians Tinggi

Patrick Amadeus Irawan - 13520109
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13520109@std.stei.itb.ac.id

Abstrak—Pengambilan keputusan yang berkaitan dengan medis wajib dilakukan dengan efektif dan dapat dipertanggungjawabkan. Salah satu metode yang dapat digunakan untuk menentukan suatu keputusan medis adalah dengan menggunakan *decision making models* dibarengi dengan sistem automasi. Penggunaan pohon keputusan merupakan salah satu metode mumpuni untuk menghasilkan klasifikasi dengan akurasi tinggi dan eksekusi cepat dengan struktur data pohon yang merepresentasikan informasi yang dapat dimanfaatkan. Akurasi ini dapat ditingkatkan dengan memanfaatkan konsep hutan acak atau penggabungan banyak pohon keputusan untuk menghindari kemungkinan kesalahan pada model dengan kompleksitas tinggi. Validitas dari metode tersebut akan dibahas secara mendalam pada paper ini.

Kata Kunci—hutan acak (*random forest*), pohon keputusan (*decision tree*), pembelajaran mesin (*machine learning*), penambangan / pengambilan data (*data mining*).

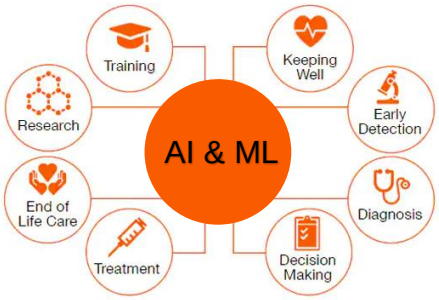
I. PENDAHULUAN

Pengambilan keputusan yang optimal merupakan kunci dari keberhasilan dalam setiap bidang. Pengambilan keputusan biasanya dibuat atas dasar kombinasi pertimbangan akan peristiwa masa lampau saat mencoba menyelesaikan permasalahan serupa baik yang berhasil maupun gagal. Di era digital ini, kebutuhan akan adanya sistem yang dapat mendukung pengambilan keputusan yang muktahir semakin meningkat dengan memproses data dengan kuantitas tinggi.

Sama pentingnya dengan bidang lain, pengambilan keputusan memegang peranan penting di bidang medis, terutama pada diagnosis individu. Sistem pendukung pengambilan keputusan yang dapat membantu tenaga medis merupakan kebutuhan yang semakin meningkat, terutama pada situasi dimana pengambilan keputusan harus dilakukan dengan cepat, tepat, dan akurat.

Pandemi COVID-19 merupakan salah satu bukti nyata dimana di berbagai negara, keputusan dan protokol medis harus dilaksanakan dengan tanggap dan efisien untuk meminimalisir hal-hal yang tidak diinginkan. Meninjau dari pertambahan populasi di Bumi dan perbandingan ketersediaan tenaga medis terhadap populasi, pada satu titik, praktik medis konvensional akan kewalahan dalam mengatasi kebutuhan medis yang kian meningkat. Atas dasar tersebut, kebutuhan terkait penerapan teknologi medis untuk meningkatkan kualtias dari pelayanan medis semakin meningkat.

Sistem pengambilan keputusan berbasis pembelajaran mesin merupakan salah satu opsi yang tengah berkembang pesat akhir-akhir ini. Didukung dengan ketersediaan data yang masif, industri kecerdasan buatan dan pembelajaran mesin dapat memajukan industri kesehatan secara menyeluruh. Mulai dari percobaan klinis, eksplorasi obat-obatan dan metode pengobatan baru, hingga invoasi perangkat medis berteknologi tinggi seperti nanopartikel, *Artifical Intelligence* dan *Machine Learning* merupakan elemen yang tidak terlepaskan berkaitan dengan transformasi digital ini.



Gambar 1.1 Penerapan AI dan ML di Bidang Kesehatan
Sumber : data-science-blog.com

Salah satu contoh algoritma Machine Learning adalah *Decision Tree Classifier* dan *Random Forest Classifier*. *Decision Tree Classifier* atau biasa dikenal sebagai Algoritma Pohon Keputusan adalah algoritma yang memanfaatkan konsep pohon sedangkan Hutan Acak adalah algoritma yang memanfaatkan konsep penggabungan keputusan dari 2 atau lebih pohon keputusan yang berbeda.

II. DASAR TEORI

A. Graf

Graf secara matematis dapat diartikan sebagai suatu struktur yang merepresentasikan objek-objek diskrit dan hubungan antara objek-objek tersebut.

Berdasarkan definisi formalnya, graf dapat dinyatakan sebagai suatu pasangan himpunan (V,E) dimana V =

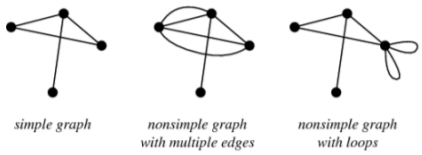
himpunan tidak kosong dari simpul-simpul (*vertices*) atau dapat dinyatakan sebagai { *v1* , *v2* , ... , *vn* } dan E = himpunan sisi (*edges*) yang menghubungkan sepasang simpul atau dapat dinyatakan sebagai {*e1* , *e2* , ... , *en* }.

Penamaan simpul pada graf dapat dituliskan dengan nomor , angka, simbol matematis dan semacamnya. Penamaan simpul dinyatakan dengan pasangan (*v1*, *v2*) atau dengan *e1*, *e2* ,... ,*en.*

Untuk membedakan jenis graf, dapat dikelompokkan baik dari aspek jumlah gelang, sisi, simpul, dan kombinasi struktural lainnya.

Berdasarkan keberadaan sisi atau gelang, graf dapat dikelompokkan menjadi :

1. Graf Sederhana
   Graf sederhana merupakan graf yang tidak mengandung gelang maupun sisi ganda.
2. Graf tak-sederhana
   Graf yang mengandung sisi ganda maupun gelang dalam strukturnya.
   2.1. Graf ganda
   Graf yang mengandung sisi ganda
   2.2. Graf semu
   Graf yang mengandung sisi berupa gelang



Gambar 2.1 Jenis graf berdasarkan keberadaan sisi atau gelang
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/

Selain itu, berdasarkan orientasi arah pada sisi graf, dapat dikelompokkan menjadi:

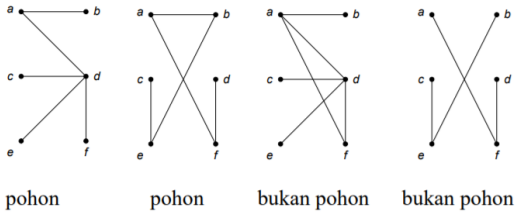
1. Graf tak-berarah
   Graf yang sisinya tidak mempunyai orientasi arah
2. Graf berarah
   Graf yang pada setiap sisinya memiliki orientasi arah, baik keluar maupun masuk (terhadap simpul)

Graf merupakan representasi struktural yang mengandung berbagai terminologi. Terminologi esensial yang berkaitan graf antara lain:

1. Ketetanggaan (*adjacent*)
   Dua buah simpul dapat dikatakan bertetangga apabila terhubung langsung
2. Bersisian (*Incidency*)
   Untuk sembarang sisi *e* = (*Vj* , *Vk)*, maka e bersisian dengan simpul Vj dan Vk.
3. Simpul Terpencil (*Isolated Vertex*)
   Simpul yang tidak mempunyai sisi yang bersisian dengan simpul itu sendiri.
4. Graf kosong
   Graf yang himpunan sisinya adalah himpunan kosong
5. Derajat
   Derajat suatu simpul adalah jumlah sisi yang bersisian dengan simpul itu sendiri.
6. Lintasan
   Panjang barisan berselang-seling sisi dan simpul yang dilalui apabila dari suatu simpul ingin menuju ke simpul lain.
7. Siklus atau Sirkuit
   Lintasan yang berawal pada simpul yang sama
8. Keterhubungan
   Dua buah simpul dikatakan terhubung jika terdapat lintasan dari simpul satu ke yang lainnya.

B. Pohon

Pohon dapat diartikan sebagai graf tak-berarah terhubung yang tidak mengandung sirkuit atau suatu struktur data yang mengandung set suatu elemen dan menyimpan representasi informasi hubungan diantara elemen-elemennya.

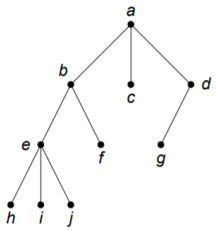


Gambar 2.2 Ilustrasi Pohon dalam bentuk graf
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/

Properti pohon dijelaskan sebagai berikut. Misalkan G = (V,E) adalah graf tak-berarah sederhana dan jumlah simpulnya sebanyak n. Pernyataan di bawah ini adalah benar :

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.

Aplikasi pohon pada umumnya menggunakan pohon dengan jenis pohon berakar. Bentuk daripada pohon berakar ini memiliki karakter unik menyerupai pohon nyata yang terbalik dimana bagian teratas merupakan akar utama dari struktur pohon diikuti upapohon lainnya yang posisinya lebih ke arah bawah.



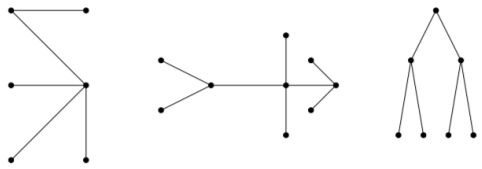
Gambar 2.3 Ilustrasi Pohon Berakar
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/

Pohon berakar merupakan representasi struktural yang mengandung berbagai terminologi. Terminologi esensial yang berkaitan dengan makalah ini antara lain:

1. Anak (*Child*)
   Simpul dapat dikatakan sebagai anak apabila terdapat sisi dari simpul tersebut ke simpul lain dengan posisi simpul yang terhubung lebih di atas.
2. Lintasan (*Path*)
   Lintasan adalah runtunan simpul-simpul yang dilalui dari suatu simpul ke simpul lainnya sedemikian sehingga setiap simpul yang dilalui merupakan *parent node* dari simpul yang berikutnya.
3. Keturunan (*descendant*) & Leluhur (*ancestor*)
   Leluhur adalah simpul dengan ketinggian yang lebih kecil dibandingkan dengan simpul terbanding, sedangka keturunan berlaku sebaliknya.
4. Upapohon (*subtree*)
   Upagraf dari pohon berakar yang mengandung simpul dan semua keturunan beserta semua sisi dalam lintasan yang berasal dari simpul berkaitan.
5. Derajat
   Jumlah anak yang dimiliki oleh simpul pohon.
6. Daun
   Simpul yang derajatnya adalah nol
7. Simpul Dalam
   Simpul yang derajatnya tidak nol
8. Tingkat (*Level*)
   Kedalaman suatu simpul relatif terhadap akar, akar memiliki tingkat 0
9. Ketinggian Pohon
   Kedalaman maksimum suatu simpul yang merupakan bagian dari pohon diukur relatif terhadap akarnya.

C. Hutan

Hutan (*forest*) diartikan sebagai sekumpulan pohon yang saling lepas atau graf tidak terhubung yang di dalamnya tidak mengandung sirkuit sama sekali.



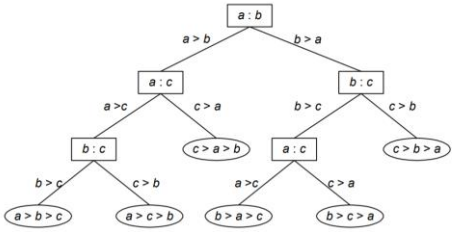
Hutan yang terdiri dari tiga buah pohon

Gambar 2.4 Ilustrasi Pohon dalam bentuk graf
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/

Untuk dengan mudah menghitung jumlah pohon yang ada di dalam suatu hutan, misalkan V adalah jumlah sisi graf dan E adalah jumlah simpul graf, kita mengetahui bahwa V - E = 1, sehingga dapat dibuktikan bahwa TV sebagai total sisi dan TE sebagai total simpul, TV – TE = Jumlah Pohon dalam Hutan.

D. Pohon Keputusan

Pohon keputusan merupakan jenis pohon yang digunakan untuk identifikasi kemungkinan pilihan dan implikasinya. Pohon keputusan terdiri atas akar dan upapohon yang bercabang terus menerus berupa pemilihan biner ataupun lebih dari dua pilihan.

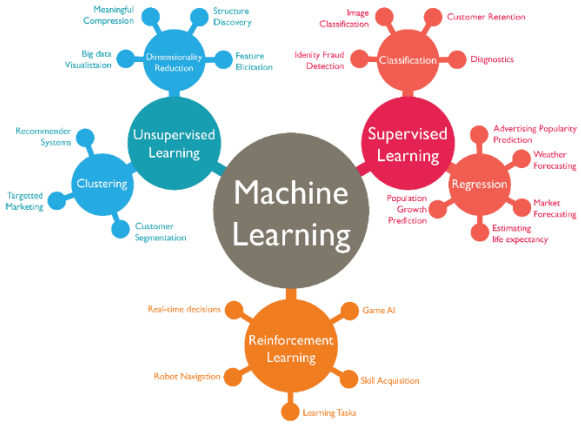


Gambar 2.5 Ilustrasi Pohon Keputusan untuk mengurutkan 3 buah elemen
Sumber : https://informatika.stei.itb.ac.id/~rinaldi.munir/

III. Analisis Aplikasi

A. *Decision Tree* dan *Random Forest* dalam *Machine Learning*

Sebelum dilakukan analisis mendalam mengenai pengaplikasian dalam bidang medis, akan dibahas terlebih dahulu aplikasi konsep pohon keputusan yang digunakan dalam pembelajaran mesin.



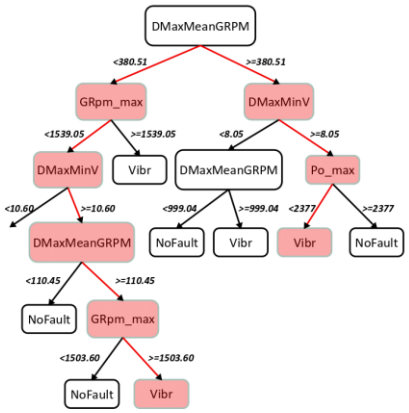
Gambar 3.1 Pengelompokan Jenis Pembelajaran Mesin
Sumber : https://7wdata.be/visualization/types-of-machine-learning-algorithms-2/

Pembelajaran mesin (*Machine Learning*) sendiri diartikan sebagai suatu cabang kecerdasan buatan yang memungkinkan mesin sebagai sistem pembelajar sehingga dapat mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data dalam waktu yang relatif singkat. Pembelajaran mesin sendiri dibagi menjadi 3 jenis, yakni:

1. Pembelajaran terarah (*supervised learning*)
   Algoritma yang mempelajari data masukan dan mengenali contoh-contoh tersebut dan membangun sebuah model yang mampu memetakan masukan baru yang belum diketahui menjadi keluaran yang akurat
2. Pembelajaran tak terarah (*unsupervised learning*)
   Algoritma yang bertujuan untuk mencari pola-pola esensial secara mandiri, meskipun tidak disediakan keluaran data secara eksplisit.
3. *Reinforcement Learning*
   Algoritma pembelajaran mesin yang mengajarkan pada kecerdasan buatan dengan pemberian *reward.*

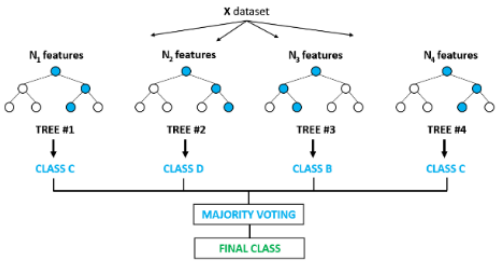
Konsep yang akan dibahas pada makalah ini tidak lain terkait pohon keputusan dan hutan acak sebagai salah satu perwujudan dari pembelajaran mesin terarah (*supervised learning*). Pada dasarnya, sistem akan menerima data masukan kemudian mengenali pola-pola yang ada di dalam data tersebut dengan menerapkan konsep pohon keputusan untuk menentukan kemungkinan keluaran yang akurat.

Pohon keputusan pada pembelajaran mesin diartikan sebagai metode pembelajaran terarah non-parametrik yang digunakan untuk klasifikasi dan regresi. Tujuan penggunaan pohon keputusan adalah untuk memberikan kesimpulan prediktif berupa keluaran terhadap masukan sembarang dengan melakukan pembelajaran aturan percabangan pohon sederhana pada data dengan jumlah besar. Perwujudan percabangan pohon pada ilustrasi di bawah ini dapat dinyatakan sebagai aproksimasi perbandingan terhadap konstanta dengan berbagai parameter pembanding.



Gambar 3.2 Ilustrasi Decision Tree Classifier
Sumber : scikit-learn.org

Sedangkan, pohon acak merupakan konsep penggunaan 2 atau lebih pohon keputusan untuk mengestimasi sampel mayoritas yang dihasilkan oleh mesin. Tujuan utama penggunaan beberapa pohon keputusan ini tidak lain untuk menghindari kesalahan prediksi apabila hanya dilakukan pengujian terhadap salah satu pohon keputusan serta meminimalisir kemungkinan *over-fitting* pada pelatihan data.



Gambar 3.3 Ilustrasi Decision Tree Classifier
Sumber : scikit-learn.org

Sesuai dengan namanya, pembelajaran mesin memerlukan modal data untuk dipelajari agar dapat membangun model yang presisi. Penggunaan algoritma pembelajaran mesin terarah, termasuk pohon keputusan, memerlukan persiapan data awal yang perlu untuk dibagi menjadi dua kelompok, yakni:

1. *Training Set*, yakni kelompok data yang dijadikan induksi atau basis pelatihan sebuah model pohon keputusan.
2. *Testing Set*, yakni kelompok data yang digunakan untuk menguji akurasi dari model yang didapat dan dapat berfungsi sebagai bahan evaluasi model.

Porsi pembagian dua kelompok tersebut juga harus disesuaikan dengan karakteristik data uji dan model algoritma yang dibentuk untuk memaksimalkan akurasi dan menghindari *overfitting* dan *underfitting*.

B. Data Mining dan *Preparation* Sampel Kasus Medis

Pembuatan suatu model evaluasi berbasis pembelajaran mesin memerlukan data uji untuk dijadikan bahan pembelajaran. Semakin banyak data yang dijadikan pembelajaran dengan kualitas yang sama atau lebih baik, tingkat akurasi model akan relatif meningkat. *Data Mining* adalah metode pengambilan data dari satu atau lebih sumber untuk dijadikan bahan baik intuk keperluan deskriptif maupun prediktif. Fungsi deskriptif ialah untuk memahami lebih lanjut karakteristik data uji dan menemukan pola bermakna, sedangkan, fungsi prediktif diartikan sebagai pencarian pola-pola pada data masa lampau untuk melakukan prediksi terhadap variabel lain yang belum diketahui nilai ataupun jenisnya.

Sampel data berkaitan dengan diagnosis kesehatan memiliki varians yang sangat tinggi, mulai dari segi statistik kuantitatif secara medis, seperti ukuran, derajat, suhu, kelembapan, luas, hingga pengaruh eksternal seperti durasi, status sosial,

Pada makalah ini, akan digunakan dataset publik yang berjudul “Breast Cancer Wisconsin (Diagnostic) Data Set” yang dikeluarkan oleh UCI Machine Learning Repository sebagai perwujudan sampel data asli yang digunakan untuk pembelajaran tingkat rendah hingga menengah karena mengandung jumlah data yang cenderung sedikit , tetapi cukup untuk dijadikan sebagai bahan model pembelajaran mesin.

```
from sklearn.datasets import load_breast_cancer
X,y = load_breast_cancer(return_X_y=True,
as frame=True)
```

Dengan menggunakan library pemrosesan pembelajaran mesin yang telah dijelaskan sebelumnya, kita dapat dengan mudah menambang data publik yang telah terdaftar pada basis data yang tersedia. Dalam proses membuat algoritma menggunakan sebuah data, ada baiknya kita memahami data yang akan kita proses terlebih dahulu dengna melakukan pemrosesan karakteristik data.

```
print(X.dtypes)
print()
print(y.dtypes)
```

```
mean radius              float64
mean texture             float64
mean perimeter           float64
mean area                float64
mean smoothness          float64
mean compactness         float64
mean concavity           float64
mean concave points      float64
mean symmetry            float64
mean fractal dimension   float64
radius error             float64
texture error            float64
perimeter error          float64
area error               float64
smoothness error         float64
compactness error        float64
concavity error          float64
concave points error     float64
symmetry error           float64
fractal dimension error  float64
worst radius             float64
worst texture            float64
worst perimeter          float64
worst area               float64
worst smoothness         float64
worst compactness        float64
worst concavity          float64
worst concave points     float64
worst symmetry           float64
worst fractal dimension  float64
dtype: object

int64
```

X merupakan representasi variabel data yang akan menentukan keluaran y. Dapat dilihat pada X, semua tipe data berbentuk bilangan real atau *float* dengan estimasi kelengkapan data 100% (dilansir dari website UCI secara langsung), ukuran X sendiri adalah 569 baris dan 30 kolom yang unik berkaitan dengan parameter medis yang menentukan jenis tumor.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.30010	0.14710	0.2419	0.07871	...
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.08690	0.07017	0.1812	0.05667	...
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.19740	0.12790	0.2069	0.05999	...
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.24140	0.10520	0.2597	0.09744	...
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.19800	0.10430	0.1809	0.05883	...
...	...	...	...	...	...	...	...	...	...	...	...
564	21.56	22.39	142.00	1479.0	0.11100	0.11590	0.24390	0.13890	0.1726	0.05623	...
565	20.13	28.25	131.20	1261.0	0.09780	0.10340	0.14400	0.09791	0.1752	0.05533	...
566	16.60	28.08	108.30	858.1	0.08455	0.10230	0.09251	0.05302	0.1590	0.05648	...
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...
568	7.76	24.54	47.92	181.0	0.05263	0.04362	0.00000	0.00000	0.1587	0.05884	...

569 rows × 30 columns

Gambar 3.4 Data parsial dari variabel X
Sumber : Dokumen Penulis

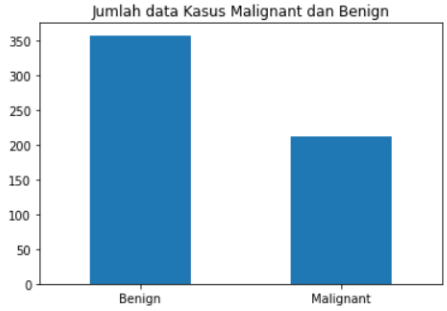
Di lain sisi, tipe data pada kolom target yang ingin dicari yaitu y merupakan bilangan bulat atau *integer*. Representasi bilangan bulat untuk kolom y berupa 1 atau 0, dengan 0 menunjukkan status kanker *malignant* (ganas) dan 1 sebagai *benign* (jinak).

```
0      0
1      0
2      0
3      0
4      0
      ..
564    0
565    0
566    0
567    0
568    1
Name: target, Length: 569, dtype: int64
```

Gambar 3.4 Data parsial dari variabel y
Sumber : Dokumen Penulis

Apabila direpresentasikan dalam bentuk grafik batang, maka distribusi data kasus jinak dan ganas terdiri atas mayoritas data dengan kasus tumor jinak. Implementasinya sebagai berikut.

```
data = y.value_counts()
data.plot(kind = 'bar')
...
print('Benign : ',data[1])
print('Malignant : ',data[0])
```



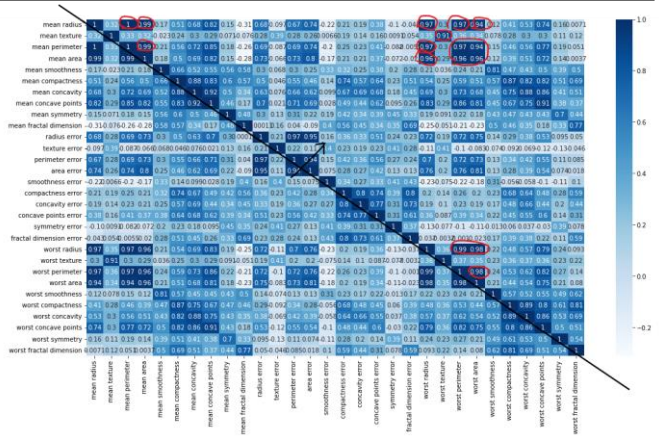

```
Benign :  357
Malignant :  212
```

Gambar 3.5 Jumlah kasus kanker ganas dan jinak
Sumber : Dokumen Penulis

Setelah kita mengenali karakteristik singkat dari data yang akan digunakan, saatnya untuk mempersiapkan data lebih lanjut dengan mengeliminasi data kotor yang berpotensi menurunkan akurasi model pohon keputusan yang akan dibuat. Data kotor mempunyai banyak jenis seperti data yang melampaui daerah nilai wajar, data yang korelasinya tidak tepat, data kosong, dan masih banyak lagi. Untuk 3 potensi data kotor pertama, sudah dipastikan data *"Breast Cancer Wisconsin (Diagnostic) Data Set"* tidak mengandung kemungkinan tersebut (dilansir dari deskripsi tautan secara langsung), untuk itu kita hanya tinggal melakukan analisis terhadap kemungkinan terakhir, yakni korelasi yang tidak sesuai.

Untuk mencari nilai korelasi secara menyeluruh, kita dapat menggunakan peta suhu (*heatmap*) untuk mengenali pasangan data dengan korelasi tidak wajar, kemudian menggunakan opsi mengganti data dengan pendekatan statistik, atau sekadar menghilangkan data tersebut untuk kasus data sederhana. Visualisasi korelasi antar variabel dapat diimplementasikan dengan bantuan library visualisasi seaborn dalam bahasa Python sebagai berikut.

```
import seaborn as sns
plt.figure(figsize=(18,10))
sns.heatmap(X.corr(), annot=True, cmap =
'Blues')
plt.show()
```



Gambar 3.6 Korelasi antar variabel di X
Sumber : Dokumen Penulis

Dari peta suhu yang terbentuk, dapat terlihat bahwa karakteristik utama yang dimiliki oleh hubungan korelasi variabel ada pada diagonalnya, yakni perbandingan terhadap variabel itu sendiri yang menghasilkan korelasi berbanding lurus, tetapi, pada bagian yang dilingkari berwarna merah, terdapat korelasi berbanding lurus antar variabel yang berbeda. Perlu dicatat bahwa untuk pembacaan peta korelasi hanya perlu membaca bagian bawah diagonal utama atau bagian atasnya.

Variabel yang saling berkorelasi tinggi ini perlu dihapus untuk menghindari permasalahan umum di pembelajaran mesin berbasis regresi, yakni permasalah multikolinear. Permasalahan multikolinear merupakan permasalahan keberadaan dua variabel bebas yang harusnya tidak berhubungan, tetapi menunjukkan hubungan kuat di dalam korelasinya. Apabila tidak ditangani, korelasi ini dapat mengganggu model pembelajaran mesin yang dibuat dan menurunkan akurasinya secara signifikan. Selain itu, menghilangkan data berarti mengurangi jumlah kompleksitas dan penyimpanan yang diperlukan untuk menjalankan suatu program sehingga memberikan manfaat tambahan. Penanganan korelasi tidak

wajar ini diimplementasikan sebagai berikut dalam bahasa Python.

```
var_drop = ['mean perimeter','mean radius','mean
compactness','mean concave points', 'radius
error', 'perimeter error', 'compactness error',
'concave points error', 'worst radius', 'worst
perimeter', 'worst compactness', 'worst concave
points', 'worst texture', 'worst area']

X = X.drop(var_drop , axis = 1)
```

Persiapan data bagian terakhir adalah membagi data menjadi dua set yang telah dibahas sebelumnya, yakni *training set* dan *testing set*. Implementasi pembagian data dengan perbandingan 4 : 1 adalah sebagai berikut.

```
from sklearn.model_selection import
train_test_split
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size = 0.2,
random_state = 42)
```

C. Aplikasi *Decision Tree* Dalam Menentukan Keputusan Medis

Tahap selanjutnya adalah pengaplikasian konsep pohon keputusan pada data yang telah siap untuk dijadikan bahan model pembelajaran mesin. Dengan menggunakan library pemrosesan scikit-learn yang telah mengandung algoritma *decision tree*, kita dapat menginisialisasi pembuatan pohon keputusan berdasarkan data yang kita miliki sembari mengatur parameter terkait untuk mendapatkan efektivitas maksimal. Implementasi pengambilan model pohon keputusan klasifikasi adalah sebagai berikut.

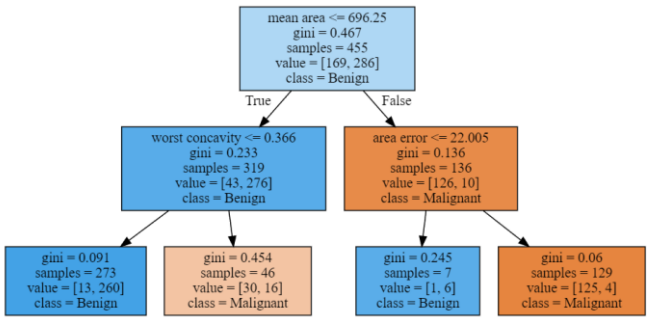
```
from sklearn.tree import DecisionTreeClassifier
tree_clf =
DecisionTreeClassifier(criterion='gini',
max_depth=2)
```

Setelah itu, kita telah membagi data menjadi 4 buah, yakni `X_test`, `X_train`, `y_test`, dan `y_train`. untuk data dengan sufiks train digunakan untuk *fitting* atau pelatihan model pohon keputusan, untuk data dengan sufiks test digunakan untuk *validation* dan *testing* model yang telah dibuat.

Menggunakan data pelatihan pada pohon keputusan dapat diimplementasikan dalam kode berikut.

```
tree_clf.fit(X_train, y_train)
y_test_pred = tree_clf.predict(X_test)
```

`tree_clf` adalah model pohon keputusan yang telah dibuat dalam kode sebelumnya dan masih kosong, untuk itu dilakukan fit dengan data training baik untuk X maupun y. `y_test_pred` menotasikan hasil prediksi dengan `X_test` yang dimiliki. Setelah melalui proses fitting, kita dapat memvisualisasikan penggamabaran pohon keputusan berdasarkan data yang telah dibuat.



Gambar 3.7 Upapohon Decision Tree classifier dari tree_clf
Sumber : Dokumen Penulis

Visualisasi tersebut adalah upapohon dari pohon keputusan secara keseluruhan yang membandingkan berbagai parameter dan menghasilkan simpul daun berupa kelas jenis kanker yakni antara `Benign` dan `Malignant`.

Tidak sampai di situ, kita dapat menguji akurasi model yang telah kita buat dengan fitur scikit-learn yakni `accuracy_score` berupa matriks penilaian. Pengujian skor ini dilakukan dengan membandingkan jawaban yang diberikan model dengan test set terhadap jawaban sebenarnya. Implementasi pembuatan matriks penilaian untuk model pohon keputusan yang telah dibuat adalah sebagai berikut.

```
from sklearn.metrics import confusion_matrix,
accuracy_score
print(f'Akurasi Model tree_clf:
{accuracy_score(y_test,y_test_pred)}')
print()
print('Matriks Penilaian')
print(f'{confusion_matrix(y_test, y_test_pred)}')
```

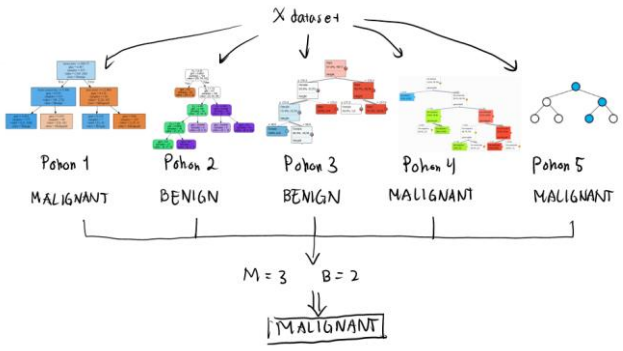
```
Output [1] :
Akurasi Model tree_clf: 0.956140350877193

Matriks Penilaian
[[40  3]
 [ 2 69]]
```

Dapat dilihat bahwa nilai yang diperoleh dari pengujian *testing data* adalah 0.95 dari 1, dimana skor tersebut termasuk sangat baik. Skor ini dapat ditingkatkan dengan menggunakan penggabungan implementasi algoritma pembelajaran mesin lain atau mengatur parameter model lama. Salah satu penggunaan algoritma yang sering digunakan bersamaan dengan pohon keputusan adalah implementasi hutan acak.

D. Aplikasi *Random Forest* dalam Meningkatkan Performa Model

Tahap terakhir adalah meningkatkan akurasi model dengan menguji algoritma baru yaitu hutan acak. Algoritma ini merupakan pengembangan sederhana dari algoritma pohon keputusan dimana digunakan banyak pohon keputusan untuk menghindari kesalahan prediktif.



Gambar 3.8 Ilustrasi Random Forest dengan n_tree = 5
Sumber : Dokumen Penulis

Ilustrasi tersebut mendeskripsikan pengambilan mayoritas keputusan yang dihasilkan oleh n buah pohon dengan fitur berbeda. Dengan sistem *voting* seperti ini, umumnya akurasi akan meningkat karena dilakukan pemeriksaan berulang kali terhadap kemungkinan keluaran yang dihasilkan pohon keputusan berbeda. Implementasi algoritma hutan acak dalam bahasa Python adalah sebagai berikut.

```
from sklearn.ensemble import
RandomForestClassifier
rf_clf = RandomForestClassifier(n_estimators=10,
max_depth=5, random_state=32, n_jobs=-1)
rf_clf.fit(X_train, y_train)
y_train_pred = rf_clf.predict(X_train)
y_test_pred = rf_clf.predict(X_test)
```

Pembuktian akurasi dapat dilakukan dengan menguji ulang matriks penilaian pada model hutan acak yang baru.

```
print(f'Akurasi data pelatihan pada Random
Forest: {accuracy_score(y_train, y_train_pred)}')
print(f'Akurasi data uji pada Random Forest:
{accuracy_score(y_test, y_test_pred)}')
print()
print('Matriks Penilaian Random Forest')
print(f'{confusion_matrix(y_test, y_test_pred)}')
```

```
Output [1] :
Akurasi data pelatihan pada Random Forest:
0.9846153846153847
Akurasi data uji pada Random Forest: 0.9649
122807017544

Matriks Penilaian Random Forest
[[41  2]
 [ 2 69]]
```

Dapat terlihat dengan jelas bahwa penilaian meningkat dari 0.95 menjadi 0.96. Pada ranah pembelajaran mesin, perbedaan persentase sebesar 1% merupakan perbedaan yang cukup signifikan sehingga penggunaan konsep hutan terbukti dapat meningkatkan performa model pembelajaran mesin yang telah dibuat.

E. Penggunaan Model secara Praktikal

Pemanfaatan praktikal dari model yang telah dibuat digunakan untuk menganalisis kasus pada pasien baru yang belum dikenali secara persis pada data-data yang ada. Dengan memanfaatkan kemampuan pengenalan pola (*pattern recognition*) dari model pohon keputusan dan hutan acak yang telah dibuat, kita dapat memprediksi kemungkinan jenis tumor yang dimiliki pasien.

Misalkan kita mempunyai data pasien yang konsultasi di rumah sakit sebagai berikut.

```
Name : John Doe
Age  : 56
Prognosis Struktural
mean texture                21.910000
mean area                 1075.000000
mean smoothness              0.094300
mean concavity               0.115300
mean symmetry                0.169200
mean fractal dimension       0.057270
texture error                1.202000
area error                  68.350000
smoothness error             0.006001
concavity error              0.028550
symmetry error               0.014920
fractal dimension error      0.002205
worst smoothness             0.146500
worst concavity              0.396500
worst symmetry               0.310900
worst fractal dimension      0.076100
```

Gambar 3.6 Data pasien baru
Sumber : Dokumen Penulis

Dengan menggunakan model hutan acak dengan akurasi tinggi yang telah dibuat, tenaga medis dapat dengan mudah menentukan kesimpulan prognosis untuk kasus dengan data seperti ilustrasi tersebut dengan implementasi berikut.

```
hasil = rf_clf.predict(np.array([2.191e+01,
1.075e+03, 9.430e-02, 1.153e-01, 1.692e-01,
5.727e-02,
       1.202e+00, 6.835e+01, 6.001e-03, 2.855e-
02, 1.492e-02, 2.205e-03,
       1.465e-01, 3.965e-01, 3.109e-01, 7.610e-
02]).reshape(1,-1))

print("Jenis Tumor pada pasien John Doe adalah" ,
'Malignant' if hasil else 'Malignant')
```

```
Output [1] :
Jenis Tumor pada pasien John Doe adalah
Malignant
```

IV. KESIMPULAN

Penerapan prinsip pohon keputusan dikombinasikan dengan algoritma hutan acak dapat menghasilkan keputusan yang akurat. Didukung dengan perkembangan teknologi komputasi pembelajaran mesin dan sumber data yang berkualitas tinggi, keputusan medis yang efektif dapat dengan mudah dihasilkan dalam waktu yang singkat.

Perwujudan kecepatan model untuk menentukan prognosis untuk kasus medis dapat menjadi modal bantuan yang besar untuk bidang kesehatan mengingat waktu merupakan aspek kritis dalam bidang berkait. Tentunya, di realita akan ada lebih banyak kasus yang lebih kompleks dan memerlukan algoritma yang lebih muktahir dan kompleks juga, dengan mengkombinasikan konsep hutan acak, pohon keputusan, dan algoritma pembelajaran mesin lainnya, dapat ditemukanalgoritma yang lebih baik dan lebih fleksibel untuk digunakan dalam berbagai kasus medis.

V. Ucapan Terima Kasih

Pertama-tama, penulis ingin menyampaikan rasa syukur terbesarnya kepada Tuhan Yang Maha Esa karena atas rahmat-Nya, makalah berjudul “Analisis Keputusan Medis Melalui Penerapan Random Forest pada Data dengan Varians Tinggi” dapat terselesaikan dengan baik dan tepat waktu. Penulis juga ingin menyampaikan rasa terima kasih kepada dosen mata kuliah IF2120 Matematika Diskrit, Dr. Ir. Rinaldi Munir, M. T., Dra.Harlili, M. Sc., dan Dr. Nur Ulfa Maulidevi, S. T, M. Sc. atas bimbingannya selama menjalani perkuliahan Matematika Diskrit sebagai bahan dalam pembuatan makalah ini. Tidak lupa penulis ingin berterimakasih kepada pihak dan sumber yang dijadikan referensi dalam pembuatan makalah ini.

Referensi

[1] https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-11-51 Diakses pada 12 Desember 2021 , 18.43 WIB.
[2] https://www.analyticsvidhya.com/blog/2021/12/application-of-tree-based-models-for-healthcare-analysis/ Diakses pada 13 Desember 2021 , 20.02 WIB.
[3] https://archive.ics.uci.edu/ml/index.php Diakses pada 13 Desember 2021 , 20.34 WIB
[4] https://informatika.stei.itb.ac.id/~rinaldi.munir/ , Diakses pada 12 Desember 2021 18.00 WIB ; 13 Desember 2021 19.54 WIB ; 14 Desember 2021 11.45 WIB
[5] https://scikit-learn.org/stable/modules/tree.html , Diakses pada 13 Desember 2021 20.59 WIB
[6] https://7wdata.be/visualization/types-of-machine-learning-algorithms-2/ , Diakses pada 14 Desember 2021 13.04 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.

Jakarta, 14 Desember 2021

Patrick Amadeus Irawan 13520109