

PROJEK AKHIR UAS
BIG DATA AND DATA MINING (ST168)

Predicting Lung Cancer Risk Using AdaBoost:
A Comprehensive Analysis of Patient Health and Environmental Factors



Disusun oleh
[22.11.5223]
[Wahyutri Nur Rohman]
[S1 Informatika 11]

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS AMIKOM YOGYAKARTA
2025

BAB I PENDAHULUAN

Kanker paru-paru merupakan salah satu jenis kanker yang paling banyak menyebabkan kematian di dunia. Berdasarkan data dari Organisasi Kesehatan Dunia (WHO), kanker paru-paru menyumbang lebih dari 1,7 juta kematian setiap tahunnya dan angka ini diperkirakan akan terus meningkat seiring dengan semakin banyaknya faktor risiko, seperti merokok, polusi udara, dan paparan bahan berbahaya di tempat kerja (WHO, 2020). Meskipun ada berbagai upaya untuk mengurangi prevalensi kanker paru-paru, deteksi dini tetap menjadi tantangan besar dalam pengobatan kanker ini. Deteksi yang terlambat sering kali mengurangi kemungkinan kesembuhan pasien, karena kanker paru-paru seringkali baru terdeteksi pada stadium lanjut.

Untuk itu, penting untuk mengembangkan sistem prediksi yang dapat memprediksi risiko kanker paru-paru lebih awal, sehingga pengobatan dapat dilakukan sebelum kanker berkembang menjadi lebih serius. Prediksi kanker paru-paru tidak hanya bergantung pada faktor medis, tetapi juga dipengaruhi oleh faktor lingkungan dan kebiasaan hidup seseorang, seperti paparan polusi udara, kebiasaan merokok, dan riwayat penyakit tertentu. Dengan demikian, analisis yang lebih mendalam dan berbasis data sangat diperlukan untuk memprediksi siapa yang berisiko tinggi terkena kanker paru-paru (Shankar & Gupta, 2019).

Teknik machine learning telah banyak digunakan untuk mengembangkan sistem prediksi kanker paru-paru berdasarkan data medis dan faktor-faktor risiko yang relevan. Salah satu metode yang menjanjikan dalam hal ini adalah AdaBoost, sebuah algoritma ensemble yang mampu mengoptimalkan model dengan menggabungkan beberapa model prediktor lemah menjadi satu model yang lebih kuat dan akurat. Beberapa penelitian sebelumnya menunjukkan bahwa AdaBoost dapat meningkatkan akurasi prediksi dalam berbagai aplikasi kesehatan, termasuk prediksi risiko kanker (Zhang & Liu, 2018). Oleh karena itu, penelitian ini bertujuan untuk menerapkan AdaBoost dalam memprediksi risiko kanker paru-paru berdasarkan data pasien yang mencakup berbagai faktor risiko.

Pada penelitian ini, digunakan dataset yang mencakup informasi penting mengenai faktor-faktor seperti usia, jenis kelamin, paparan polusi udara, kebiasaan merokok, serta gejala klinis seperti batuk darah, nyeri dada, dan sesak napas. Setiap faktor ini dapat mempengaruhi kemungkinan seseorang mengembangkan kanker paru-paru. Dalam proses analisis, dataset ini akan diproses terlebih dahulu melalui tahapan pembersihan data dan seleksi fitur yang relevan untuk mengurangi noise dan meningkatkan kualitas model. Selanjutnya, model AdaBoost akan diterapkan untuk menghasilkan prediksi yang akurat mengenai kemungkinan seseorang mengembangkan kanker paru-paru.

Tujuan utama dari penelitian ini adalah untuk membangun model prediksi risiko kanker paru-paru yang dapat memberikan hasil yang lebih akurat dengan memanfaatkan algoritma AdaBoost. Diharapkan model ini bisa digunakan oleh tenaga medis sebagai alat bantu untuk deteksi dini, sehingga pasien yang berisiko tinggi dapat segera menerima perawatan yang lebih intensif. Dengan

memanfaatkan algoritma machine learning seperti AdaBoost, diharapkan sistem prediksi ini dapat mengurangi keterlambatan diagnosis yang sering terjadi pada kanker paru-paru dan memberikan harapan lebih bagi pasien untuk mendapatkan pengobatan yang efektif (Kumar & Vohra, 2020).

BAB 2 PROFIL DATASET

a) **Karakteristik Data**

Dataset ini berisi informasi tentang pasien yang terkait dengan kanker paru-paru dan berbagai faktor risiko yang mungkin berkontribusi pada perkembangan penyakit tersebut. Berikut adalah deskripsi lengkap dari kolom-kolom dalam dataset ini:

1. **index**: Indeks atau ID unik untuk setiap baris data.
2. **Patient Id**: ID pasien yang juga dapat digunakan untuk mengidentifikasi individu dalam dataset.
3. **Age**: Usia pasien, yang merupakan faktor penting dalam risiko kanker paru-paru. Usia yang lebih tua dapat meningkatkan kemungkinan terkena kanker paru-paru.
4. **Gender**: Jenis kelamin pasien, yang bisa mempengaruhi prevalensi kanker paru-paru.
5. **Air Pollution**: Tingkat paparan polusi udara yang merupakan faktor risiko lingkungan yang signifikan dalam perkembangan kanker paru-paru.
6. **Alcohol use**: Penggunaan alkohol, yang dapat menjadi faktor risiko kanker paru-paru.
7. **Dust Allergy**: Riwayat alergi debu, yang bisa berhubungan dengan sensitivitas terhadap polusi udara dan penyakit paru-paru.
8. **Occupational Hazards**: Paparan terhadap bahaya lingkungan di tempat kerja, seperti bahan kimia berbahaya atau debu yang dapat berkontribusi pada kanker paru-paru.
9. **Genetic Risk**: Risiko genetik berdasarkan riwayat keluarga atau faktor keturunan yang bisa berpengaruh pada perkembangan kanker paru-paru.
10. **Chronic Lung Disease**: Riwayat penyakit paru-paru kronis yang meningkatkan risiko kanker paru-paru.
11. **Balanced Diet**: Status pola makan seimbang, yang memengaruhi kesehatan umum tubuh dan ketahanan terhadap penyakit.
12. **Obesity**: Status obesitas berdasarkan Indeks Massa Tubuh (BMI), yang dapat berhubungan dengan peningkatan risiko beberapa jenis kanker.
13. **Smoking**: Kebiasaan merokok, yang merupakan faktor utama penyebab kanker paru-paru.
14. **Passive Smoker**: Paparan terhadap asap rokok orang lain, yang juga berisiko meningkatkan kemungkinan kanker paru-paru.

15. **Chest Pain:** Gejala nyeri dada yang dapat menjadi tanda awal masalah pernapasan atau kanker paru-paru.
16. **Coughing of Blood:** Batuk dengan darah, yang seringkali merupakan gejala dari kanker paru-paru.
17. **Fatigue:** Kelelahan yang tidak biasa, yang seringkali terjadi pada pasien kanker paru-paru.
18. **Weight Loss:** Penurunan berat badan yang tidak dapat dijelaskan, gejala umum pada banyak jenis kanker termasuk kanker paru-paru.
19. **Shortness of Breath:** Sesak napas yang sering terjadi pada kanker paru-paru dan penyakit paru-paru lainnya.
20. **Wheezing:** Dengkuran atau suara napas yang berbunyi, dapat menjadi gejala dari penyakit paru-paru.
21. **Swallowing Difficulty:** Kesulitan menelan yang dapat mengindikasikan kanker paru-paru atau masalah lain pada sistem pernapasan.
22. **Clubbing of Finger Nails:** Perubahan pada kuku jari, yang sering kali terjadi pada pasien dengan kanker paru-paru stadium lanjut.
23. **Frequent Cold:** Sering flu, yang bisa menjadi tanda dari gangguan sistem kekebalan tubuh.
24. **Dry Cough:** Batuk kering, gejala yang sering terlihat pada kanker paru-paru.
25. **Snoring:** Mendengkur, yang meskipun lebih umum terkait dengan gangguan tidur, juga bisa terkait dengan masalah pernapasan.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 26 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   index                                1000 non-null   int64
1   Patient Id                           1000 non-null   object
2   Age                                   1000 non-null   int64
3   Gender                               1000 non-null   int64
4   Air Pollution                         1000 non-null   int64
5   Alcohol use                           1000 non-null   int64
6   Dust Allergy                         1000 non-null   int64
7   Occupational Hazards                  1000 non-null   int64
8   Genetic Risk                         1000 non-null   int64
9   Chronic Lung Disease                  1000 non-null   int64
10  Balanced Diet                         1000 non-null   int64
11  Obesity                               1000 non-null   int64
12  Smoking                               1000 non-null   int64
13  Passive Smoker                        1000 non-null   int64
14  Chest Pain                           1000 non-null   int64
15  Coughing of Blood                     1000 non-null   int64
16  Fatigue                               1000 non-null   int64
17  Weight Loss                           1000 non-null   int64
18  Shortness of Breath                   1000 non-null   int64
19  Wheezing                              1000 non-null   int64
20  Swallowing Difficulty                 1000 non-null   int64
21  Clubbing of Finger Nails              1000 non-null   int64
22  Frequent Cold                         1000 non-null   int64
23  Dry Cough                             1000 non-null   int64
24  Snoring                               1000 non-null   int64
25  Level                                 1000 non-null   object
dtypes: int64(24), object(2)
memory usage: 203.3+ KB
```

Gambar.1 Isi Dataset

Target Variabel:

- **Level:** Tingkat keparahan kanker atau apakah pasien terdiagnosis kanker paru-paru. Variabel ini digunakan untuk memprediksi apakah seseorang memiliki risiko tinggi terkena kanker paru-paru berdasarkan faktor-faktor yang tercantum di atas.

Kualitas Data:

- **Missing Values:** Dataset ini mengandung nilai yang hilang pada beberapa fitur, terutama fitur numerik. Pengolahan yang tepat seperti imputasi nilai rata-rata atau modus untuk kategori diperlukan untuk menangani missing values.
- **Data Kategorikal dan Numerik:** Dataset ini terdiri dari fitur numerik (seperti usia, air pollution, dan BMI) serta data kategorikal (seperti gender, smoking, dan gejala). Pengolahan yang tepat harus dilakukan, seperti encoding untuk data kategorikal agar bisa digunakan dalam model machine learning.
- **Keseimbangan Kelas:** Variabel target **Level** mungkin memiliki distribusi yang tidak seimbang antara kategori "terkena kanker" dan "tidak terkena kanker," yang bisa

mempengaruhi akurasi model. Teknik seperti oversampling atau undersampling dapat digunakan untuk mengatasi masalah ketidakseimbangan kelas.

- b) **Sumber Data:** Dataset ini diambil dari Kaggle dengan nama "Cancer Patients and Air Pollution: A New Link". Dataset ini menghubungkan data tentang pasien kanker dengan paparan polusi udara, untuk menganalisis kemungkinan keterkaitannya dengan kanker paru-paru.

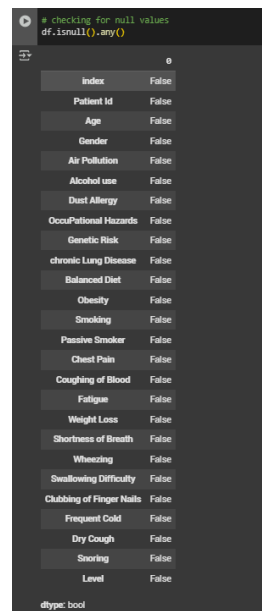
Link ke Dataset: <https://www.kaggle.com/datasets/thedevastator/cancer-patients-and-air-pollution-a-new-link/data>

BAB 3 PREPROSESING DATA

Preprocessing data adalah proses persiapan data mentah sebelum digunakan untuk analisis atau pelatihan model machine learning. Ibarat memasak, preprocessing adalah langkah membersihkan dan memotong bahan-bahan sebelum dimasak. Proses ini penting karena data mentah seringkali tidak langsung siap dipakai. Dimana pada project yang saya kerjakan ini menggunakan beberapa macam teknik seperti dibawah ini.

1. pengecekan missing value.

Penanganan missing value dilakukan untuk mengatasi data yang kosong dalam dataset. Langkah ini penting karena data yang kosong dapat mengganggu proses analisis atau pelatihan model machine learning. Untuk mengatasinya, nilai kosong dapat diisi menggunakan rata-rata, median, atau modus dari data lainnya. Jika data kosong terlalu banyak atau tidak relevan, baris atau kolom tersebut dapat dihapus agar dataset lebih bersih dan konsisten.



```
# checking for null values
df.isnull().any()
```

	0
Index	False
Patient Id	False
Age	False
Gender	False
Air Pollution	False
Alcohol use	False
Dust Allergy	False
Occupational Hazards	False
Genetic Risk	False
chronic Lung Disease	False
Balanced Diet	False
Obesity	False
Smoking	False
Passive Smoker	False
Chest Pain	False
Coughing of Blood	False
Fatigue	False
Weight Loss	False
Shortness of Breath	False
Wheezing	False
Swallowing Difficulty	False
Clubbing of Finger Nails	False
Frequent Cold	False
Dry Cough	False
Sneezing	False
Level	False

dtype: bool

gambar 2. mengecek missing values.

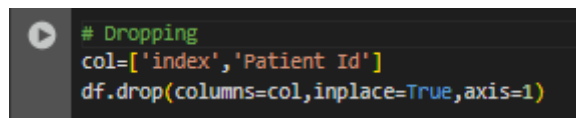
Pada gambar yang disebutkan, **mengecek missing values** adalah proses untuk memastikan apakah ada data yang hilang atau kosong dalam dataset yang digunakan. Missing values dapat menyebabkan masalah dalam analisis dan pelatihan model machine learning, karena kebanyakan algoritma membutuhkan data yang lengkap untuk menghasilkan hasil yang akurat. Oleh karena itu, penting untuk memeriksa apakah ada nilai yang kosong dalam dataset sebelum melanjutkan ke langkah berikutnya.

Namun, dalam data yang digunakan, hasil pengecekan menunjukkan bahwa tidak ada missing values atau nilai yang kosong. Ini berarti bahwa setiap kolom dalam dataset memiliki data yang lengkap untuk setiap baris. Hal ini sangat menguntungkan karena tidak perlu lagi

melakukan penanganan khusus terhadap data yang hilang, seperti mengisi nilai yang kosong dengan nilai rata-rata, median, atau modus, atau bahkan menghapus baris yang memiliki missing values.

2. Penghapusan kolom yang tidak dipakai.

Penghapusan kolom yang tidak dipakai bertujuan untuk menyederhanakan dataset dengan menghapus informasi yang tidak relevan terhadap analisis atau prediksi. Contohnya, kolom seperti "Nama" atau "ID" biasanya tidak memberikan kontribusi langsung terhadap hasil dan dapat diabaikan. Langkah ini membantu meningkatkan efisiensi pemrosesan data sekaligus mengurangi kompleksitas model.

A screenshot of a code editor showing Python code to drop columns. The code is:

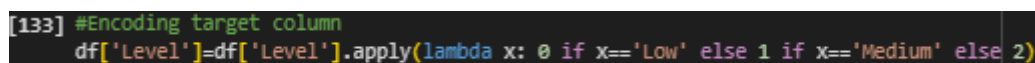
```
# Dropping  
col=['index','Patient Id']  
df.drop(columns=col,inplace=True,axis=1)
```

Gambar 3. menghapus kolom yang tidak digunakan.

Kolom seperti **index** dan **patient ID** dihapus karena tidak memiliki relevansi langsung terhadap analisis atau prediksi. Kolom ini hanya berfungsi sebagai identifikasi unik untuk setiap baris data dan tidak memberikan informasi bermakna yang dapat membantu model machine learning. Jika dibiarkan, model dapat salah menginterpretasikan kolom tersebut sebagai fitur penting, padahal hanya berupa angka acak yang tidak memiliki hubungan dengan target. Selain itu, menghapus kolom yang tidak relevan dapat menyederhanakan dataset, sehingga proses analisis menjadi lebih efisien dan fokus hanya pada fitur-fitur yang benar-benar memberikan kontribusi terhadap hasil. Oleh karena itu, penghapusan kolom seperti **index** dan **patient ID** adalah langkah penting untuk memastikan bahwa data yang digunakan hanya terdiri dari informasi yang berguna.

3. Encoding label pada columns yang digunakan sebagai column target.

Encoding label pada kolom target dilakukan untuk mengubah data kategorikal menjadi angka yang dapat dipahami oleh model machine learning.

A screenshot of a code editor showing Python code to encode a target column. The code is:

```
[133] #Encoding target column  
df['Level']=df['Level'].apply(lambda x: 0 if x=='Low' else 1 if x=='Medium' else 2)
```

Gambar 4. Melakukan encoding label.

Digunakan untuk mengubah nilai-nilai dalam kolom **Level** yang bersifat kategorikal menjadi nilai numerik agar dapat diproses oleh model machine learning. Kolom **Level** berisi

kategori seperti 'Low', 'Medium', dan 'High', yang diubah menjadi angka 0, 1, dan 2 menggunakan fungsi **lambda**. Proses ini penting karena sebagian besar algoritma machine learning hanya dapat menangani data numerik, bukan data teks atau kategori. Dengan mengganti kategori tersebut menjadi angka, kita memudahkan model untuk mempelajari pola dan hubungan yang ada dalam data. Langkah ini juga membantu dalam membuat dataset lebih efisien dan meminimalisir potensi kesalahan dalam proses pelatihan model.

4. Normalisasi data.

Normalisasi data bertujuan untuk menyetarakan skala semua fitur dalam dataset, terutama jika terdapat perbedaan rentang nilai yang besar. Misalnya, nilai 1 juta hingga 100 juta dapat dinormalisasi ke rentang 0-1. Hal ini membantu model bekerja lebih stabil, cepat, dan akurat karena fitur-fitur dalam dataset berada pada skala yang sama.

```
Kode yang dihasilkan mungkin tunduk pada lisensi |
#lakukan normalisasi data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
```

Gambar 5. melakukan normalisasi data.

Pada **Gambar 5**, yang menunjukkan **proses normalisasi data**, kode yang digunakan bertujuan untuk mengubah skala nilai-nilai fitur dalam dataset agar berada dalam rentang yang seragam. Proses normalisasi sangat penting terutama ketika dataset mengandung fitur dengan rentang nilai yang sangat berbeda-beda. Misalnya, jika salah satu kolom berisi nilai dalam rentang 0 hingga 1, sementara kolom lain memiliki nilai dari ribuan hingga jutaan, perbedaan skala ini bisa memengaruhi kinerja model machine learning.

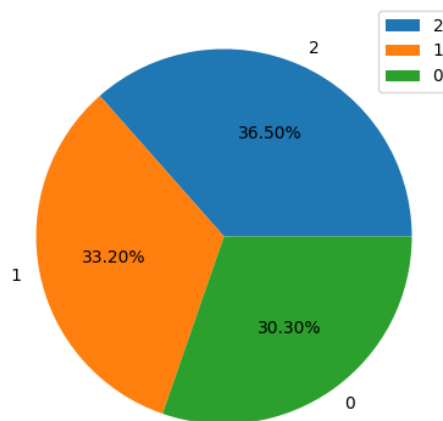
BAB 4 EDA

Exploratory Data Analysis (EDA) adalah proses analisis awal pada dataset untuk memahami karakteristik, pola, dan hubungan antar fitur. EDA bertujuan untuk menggali wawasan dari data dan menemukan potensi masalah, seperti missing values, outliers, atau distribusi data yang tidak normal, yang dapat memengaruhi hasil analisis atau model. Proses ini mencakup berbagai langkah, seperti memeriksa ukuran dataset, jenis data pada setiap kolom, serta nilai-nilai unik. Visualisasi juga menjadi bagian penting dari EDA, misalnya dengan menggunakan histogram untuk melihat distribusi data, scatter plot untuk hubungan antar variabel, atau heatmap untuk menganalisis korelasi.

Selain itu, langkah EDA membantu mengidentifikasi fitur yang paling relevan untuk target, menemukan pola outlier, serta mengevaluasi apakah dataset sudah seimbang. EDA merupakan langkah penting untuk memastikan data siap digunakan dan menghasilkan keputusan yang lebih baik dalam proses analitik atau pemodelan machine learning. Dimana dalam pembuatan project ini juga melakukan explorasi data sebagai berikut.

1. Pie Chart pada Kolom Target 'Level'.

Pie chart digunakan untuk menunjukkan proporsi masing-masing kategori dalam kolom target Level (misalnya, "Low", "Medium", dan "High"). Ini memberikan gambaran visual yang mudah dipahami tentang bagaimana distribusi target di dataset, apakah seimbang atau ada ketidakseimbangan antara kategori-kategori tersebut. Jika distribusinya tidak seimbang, kita mungkin perlu menggunakan teknik khusus dalam pemodelan untuk menangani ketidakseimbangan tersebut.

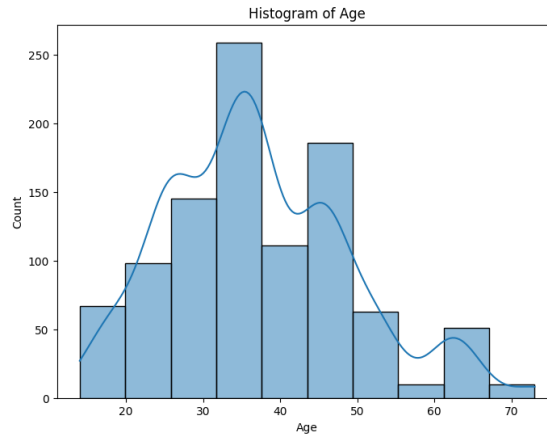


Gambar 6. Pie chart level.

Dalam gambar di atas, dapat terlihat bahwa presentasi kategori tertinggi pada kolom Level adalah "High" dengan persentase sebesar 36,50%, diikuti oleh "Medium" sebesar 33,20%, dan "Low" sebesar 30,30%. Hal ini menunjukkan bahwa meskipun terdapat ketiga kategori, distribusinya relatif seimbang, dengan sedikit dominasi pada kategori "High".

2. Histogram of Age.

Histogram digunakan untuk melihat distribusi usia dalam dataset. Ini membantu dalam memahami apakah data usia tersebar secara merata atau ada pola tertentu, seperti usia yang lebih dominan. Histogram juga bisa mengungkapkan adanya outlier atau nilai ekstrem yang mungkin perlu ditangani.

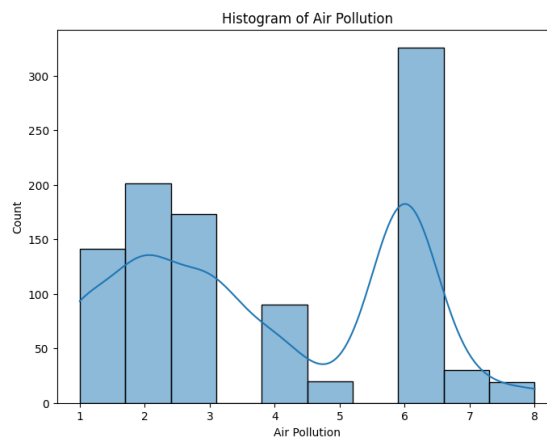


Gambar 7. Histogram umur.

Pada gambar di atas, dapat terlihat bahwa distribusi usia pada kolom Age menunjukkan bahwa sebagian besar penderita penyakit paru-paru berada pada kisaran usia 30-40 tahun. Hal ini terlihat dari puncak histogram yang menunjukkan konsentrasi tertinggi pada rentang usia tersebut. Informasi ini sangat penting karena memberikan wawasan tentang kelompok usia yang lebih rentan terhadap penyakit paru-paru dalam dataset. Dengan pemahaman ini, analisis lebih lanjut dapat dilakukan untuk mengeksplorasi faktor-faktor lain yang mungkin berkontribusi pada prevalensi penyakit tersebut di kelompok usia ini, serta untuk merencanakan strategi pencegahan atau pengobatan yang lebih efektif untuk kelompok tersebut.

3. Histogram of Air Pollution.

Menampilkan histogram untuk Air Pollution bertujuan untuk menganalisis bagaimana tingkat polusi udara tersebar di seluruh dataset. Ini bisa membantu kita melihat apakah sebagian besar data memiliki tingkat polusi udara rendah atau tinggi, serta mengidentifikasi jika ada outlier yang perlu diperhatikan atau dikoreksi.

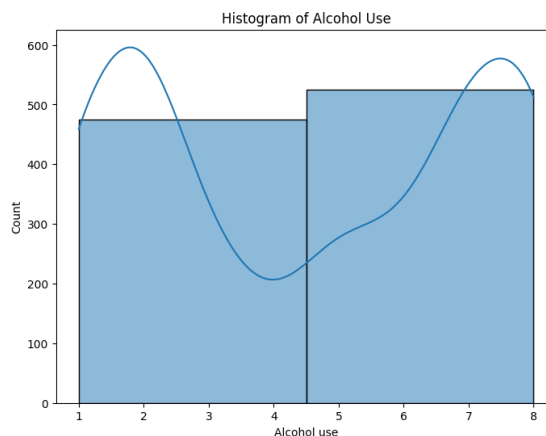


Gambar 8. Histogram polusi udara.

Pada gambar di atas, dapat dilihat bahwa tingkat tertinggi penyebab penyakit paru-paru dalam dataset ini terkait dengan polusi udara. Hal ini terlihat dari histogram Air Pollution, di mana mayoritas individu dalam dataset terpapar pada tingkat polusi udara yang tinggi. Polusi udara yang tinggi dapat menjadi faktor risiko utama bagi kesehatan pernapasan, dan dalam konteks ini, dapat berkontribusi pada tingginya prevalensi penyakit paru-paru. Dengan informasi ini, kita bisa mengidentifikasi polusi udara sebagai faktor penting yang perlu mendapat perhatian dalam upaya pencegahan dan penanggulangan penyakit paru-paru, serta pentingnya kebijakan lingkungan untuk mengurangi paparan polusi udara bagi masyarakat.

4. **Histogram of Alcohol Use.**

Dengan menampilkan histogram Alcohol Use, kita bisa mengetahui pola konsumsi alkohol dalam dataset, apakah konsumsi alkohol sebagian besar berada pada tingkat rendah, sedang, atau tinggi. Ini juga memberikan wawasan tentang distribusi data dan apakah ada bias atau ketidakseimbangan dalam kategori tersebut.

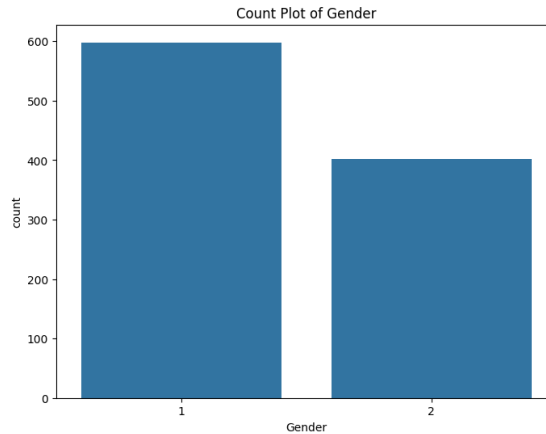


Gambar 9. Histogram konsumsi alkohol.

Pada gambar di atas, histogram Alcohol use menunjukkan distribusi tingkat konsumsi alkohol dalam dataset. Meskipun konsumsi alkohol tampaknya terbagi menjadi dua kategori (misalnya, "Tinggi" dan "Rendah"), penting untuk dicatat bahwa tingkat konsumsi alkohol yang tinggi dapat mempengaruhi kesehatan paru-paru. Konsumsi alkohol yang berlebihan sering dikaitkan dengan berbagai masalah kesehatan, termasuk gangguan pada sistem pernapasan. Meskipun dalam gambar ini data menunjukkan distribusi konsumsi alkohol, hubungan langsung dengan kondisi paru-paru memerlukan analisis lebih lanjut. Jika ada pola yang menunjukkan prevalensi lebih tinggi pada individu dengan konsumsi alkohol tinggi, maka ini dapat menjadi faktor risiko yang signifikan dalam perkembangan penyakit paru-paru.

5. **Count Plot of Gender.**

Count plot digunakan untuk menampilkan jumlah data berdasarkan kategori Gender (misalnya, Laki-laki dan Perempuan). Ini memberikan gambaran tentang bagaimana distribusi gender di dataset dan apakah ada perbedaan signifikan antara kategori tersebut yang bisa mempengaruhi hasil analisis.

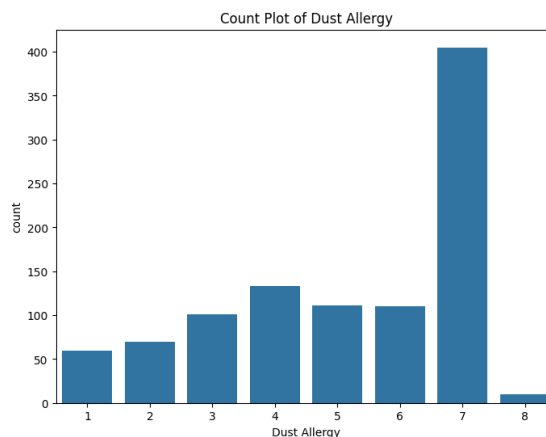


Gambar 10. Plot jenis kelamin.

Pada gambar di atas, count plot Gender menunjukkan bahwa kebanyakan penderita penyakit paru-paru dalam dataset ini adalah laki-laki. Hal ini terlihat dari jumlah individu laki-laki yang lebih banyak dibandingkan dengan perempuan dalam plot tersebut. Informasi ini penting karena menunjukkan adanya kemungkinan kecenderungan atau faktor risiko yang lebih tinggi pada laki-laki terkait dengan penyakit paru-paru dalam dataset yang digunakan. Meskipun begitu, penting untuk melakukan analisis lebih lanjut untuk memahami faktor-faktor lain yang mungkin berperan, seperti gaya hidup, kebiasaan merokok, atau paparan polusi udara, yang juga dapat mempengaruhi prevalensi penyakit berdasarkan jenis kelamin.

6. Count Plot of Dust Allergy.

Visualisasi count plot untuk Dust Allergy memberikan informasi tentang jumlah orang yang memiliki alergi debu dan yang tidak. Ini membantu memahami prevalensi kondisi ini dalam dataset dan apakah faktor ini memiliki hubungan yang signifikan dengan target atau fitur lainnya.

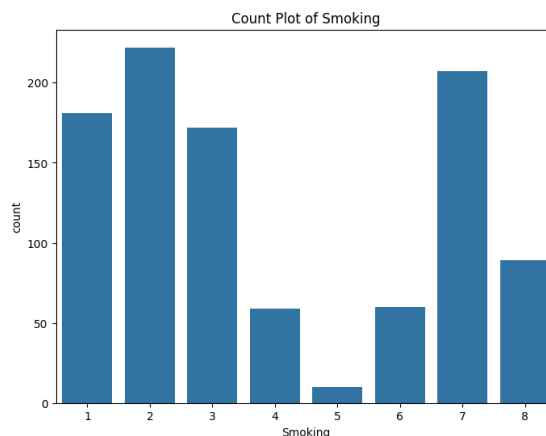


Gambar 11. Plot alergi debu.

Pada gambar di atas, count plot Dust Allergy menunjukkan bahwa kebanyakan penderita penyakit paru-paru dalam dataset ini memiliki riwayat alergi terhadap debu. Hal ini terlihat dari jumlah individu dengan alergi debu yang lebih dominan dibandingkan mereka yang tidak memiliki alergi tersebut. Alergi debu dapat menjadi salah satu faktor pemicu yang signifikan, karena paparan debu dapat memperburuk kondisi pernapasan dan memicu berbagai masalah kesehatan, termasuk penyakit paru-paru. Informasi ini menunjukkan pentingnya mengidentifikasi dan mengelola alergi debu sebagai langkah preventif untuk mengurangi risiko atau dampak penyakit paru-paru, terutama pada individu yang rentan.

7. Count Plot of Smoking.

Count plot untuk Smoking menunjukkan seberapa banyak individu yang merokok dan tidak merokok dalam dataset. Ini membantu dalam melihat pola kebiasaan merokok dan apakah ada hubungan antara kebiasaan merokok dan kondisi kesehatan tertentu yang tercatat dalam dataset.

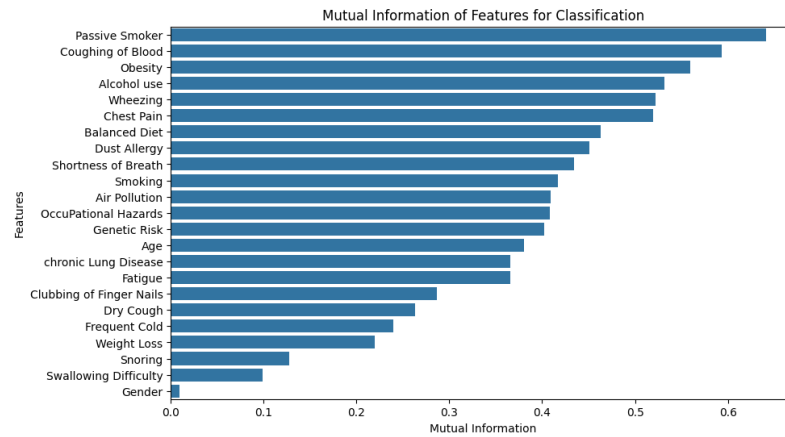


Gambar 12. Plot penggunaan rokok.

Pada gambar di atas, count plot Smoking menunjukkan bahwa tingkat kebiasaan merokok tertinggi berada pada kategori 2 dan 7 (yang mungkin merepresentasikan skala tertentu dari tingkat kebiasaan merokok, seperti intensitas atau durasi). Hal ini menunjukkan bahwa sebagian besar penderita dalam dataset memiliki tingkat kebiasaan merokok yang cukup signifikan pada kedua kategori tersebut. Merokok merupakan salah satu faktor risiko utama untuk berbagai masalah kesehatan, termasuk penyakit paru-paru, karena dapat merusak jaringan paru-paru dan mengurangi kapasitas pernapasan. Data ini menegaskan perlunya perhatian khusus terhadap kebiasaan merokok, terutama pada individu dengan kategori tertinggi, untuk mencegah dan mengurangi prevalensi penyakit paru-paru.

8. Plotting Mutual Information.

Plot mutual information membantu kita memahami hubungan antara fitur-fitur dalam dataset dengan target variabel. Ini memberikan informasi tentang seberapa besar keterkaitan masing-masing fitur dengan target, baik dalam hubungan linear maupun non-linear. Fitur dengan nilai mutual information tinggi akan lebih berpengaruh terhadap prediksi model, sementara yang rendah bisa dianggap kurang penting.



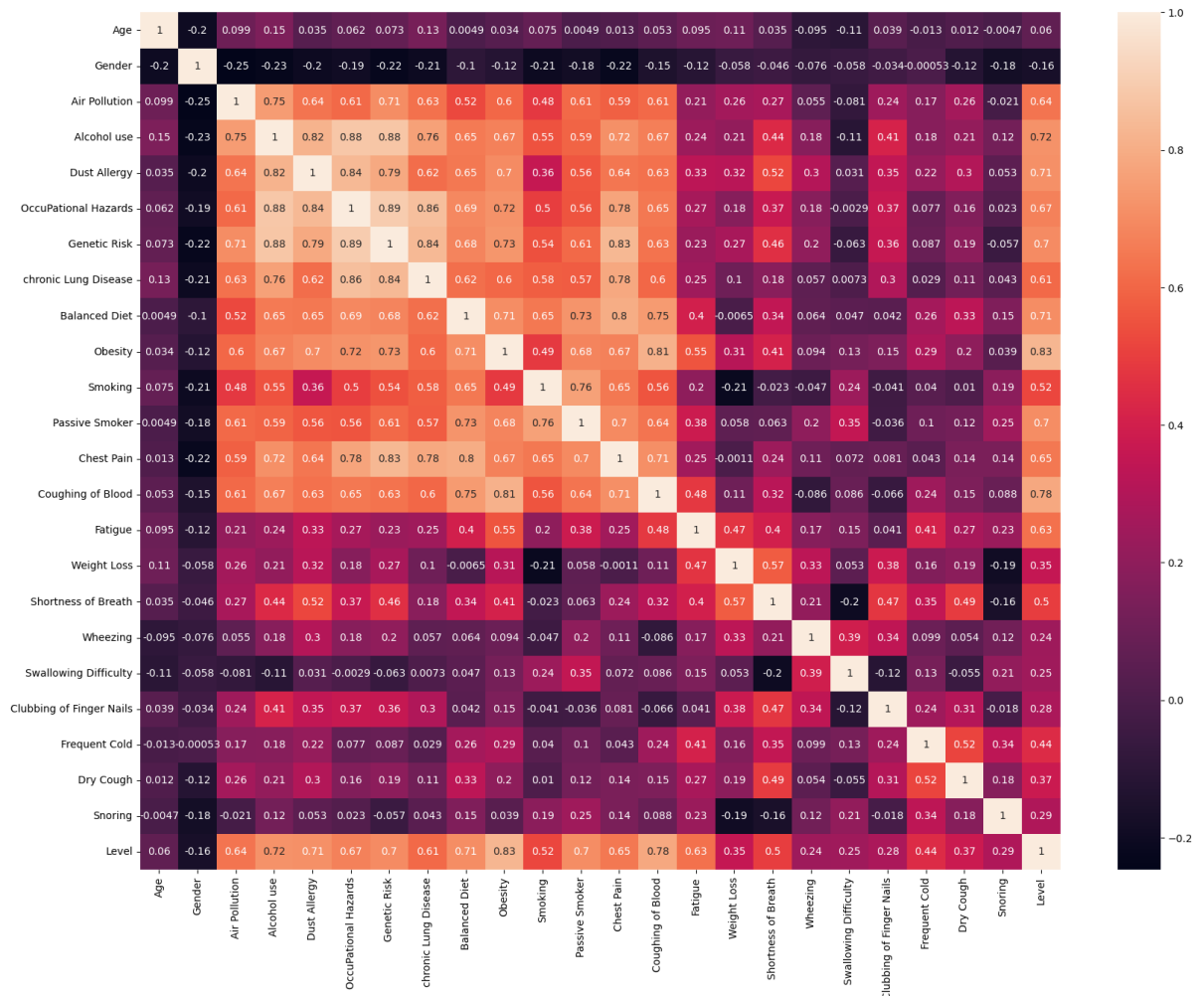
Gambar 13. Mutual Informasi.

Pada plot Mutual Information, terlihat bahwa tiga fitur dengan nilai mutual information tertinggi terhadap kolom target adalah passive smoker, coughing of blood, dan obesity. Hal ini menunjukkan bahwa ketiga fitur tersebut memiliki hubungan yang paling kuat, baik secara linear maupun non-linear, dengan target variabel (misalnya, tingkat keparahan atau keberadaan penyakit paru-paru).

1. **Passive Smoker:** Paparan terhadap asap rokok secara pasif sering kali memiliki dampak kesehatan yang hampir sama buruknya dengan merokok aktif. Dalam konteks ini, nilai mutual information yang tinggi menunjukkan bahwa menjadi perokok pasif adalah salah satu faktor yang paling signifikan terkait dengan penyakit paru-paru dalam dataset.
2. **Coughing of Blood:** Gejala batuk darah merupakan indikator klinis yang kuat dari masalah serius pada sistem pernapasan, seperti infeksi, kerusakan jaringan, atau penyakit kronis lainnya. Tingginya nilai mutual information menegaskan bahwa gejala ini sangat relevan dalam menentukan atau memprediksi kondisi target.
3. **Obesity:** Obesitas sering kali dikaitkan dengan gangguan pada sistem pernapasan, seperti sleep apnea atau sesak napas, yang dapat memperburuk kondisi paru-paru. Hubungan yang signifikan antara obesitas dan target variabel menunjukkan bahwa berat badan yang berlebih adalah salah satu faktor penting dalam analisis ini.

9. Corelation

Korelasi adalah ukuran statistik yang digunakan untuk menunjukkan hubungan antara dua variabel, baik dari segi arah maupun kekuatannya. Korelasi membantu kita memahami sejauh mana satu variabel berubah seiring dengan perubahan variabel lainnya. Hubungan ini dinyatakan dalam nilai koefisien korelasi yang berkisar antara -1 hingga 1.



Gambar 14. Korelasi antar fitur

Pada gambar diatas dari hasil korelasi yang ditampilkan pada heatmap di atas, fitur-fitur yang memiliki korelasi sangat tinggi terhadap target variabel (Level) perlu diidentifikasi untuk membantu proses seleksi fitur. Berikut langkah-langkah yang dapat dilakukan untuk memilih fitur:

1. Identifikasi Korelasi yang Tinggi dengan Target (Level) Fokus pada fitur-fitur dengan nilai korelasi yang mendekati 1 (positif kuat) atau -1 (negatif kuat) terhadap kolom target Level.
2. Threshold Korelasi tetapkan ambang batas untuk korelasi, misalnya hanya memilih fitur dengan korelasi di atas 0.6 atau 0.7 dengan target variabel.

3. Hindari Multikolinearitas jika dua fitur memiliki korelasi tinggi satu sama lain (misalnya di atas 0.8), pilih salah satu fitur yang lebih relevan untuk menghindari redundansi.

Contoh Fitur dengan Korelasi Tinggi Dari heatmap, terlihat fitur seperti:

- Obesity (korelasi ~0.83 dengan Level),
- Coughing of Blood (~0.78),
- Passive Smoker (~0.76),
- Air Pollution (~0.72),
- Alcohol Use (~0.71),
- Balanced Diet (~0.71),
- chronic Lung Disease (~0.76),
- Genetic Risk (~0.7).

BAB 5 SELEKSI FITUR

Jika hasil akurasi menurun setelah dilakukan seleksi fitur, maka keputusan untuk tidak melakukan seleksi fitur dapat dibenarkan. Hal ini disebabkan oleh beberapa alasan yang relevan:

1. Kehilangan Informasi Penting kasus, meskipun fitur tertentu memiliki korelasi rendah dengan target, fitur tersebut dapat berkontribusi pada model secara tidak langsung (misalnya, melalui interaksi dengan fitur lain). Dengan menghapusnya, informasi penting mungkin hilang, yang pada akhirnya memengaruhi performa model.
2. Pengaruh Fitur Minor pada Akurasi, Beberapa fitur yang tampaknya tidak terlalu penting mungkin tetap memiliki pengaruh kecil yang, jika digabungkan, dapat meningkatkan akurasi model secara keseluruhan.
3. Tujuan Akhir Adalah Akurasi, Jika tujuan utama proyek Anda adalah mencapai akurasi setinggi mungkin, maka mempertahankan semua fitur, bahkan yang kurang signifikan, adalah langkah yang tepat selama tidak ada kendala sumber daya komputasi.

```
import pandas as pd

# Hitung korelasi
correlation_matrix = df.corr()

# Tentukan ambang batas korelasi tinggi (threshold)
threshold = 0.7

# Pilih fitur dengan korelasi tinggi terhadap target 'Level' (kecuali target itu sendiri)
high_corr_features = correlation_matrix['Level'].abs() > threshold].index.tolist()
high_corr_features.remove('Level') # Hilangkan target dari daftar fitur

# Definisikan X (fitur) dan Y (target)
X = df[high_corr_features]
Y = df['Level']

# Tampilkan fitur yang dipilih
print("Fitur dengan korelasi tinggi terhadap Level:")
print(high_corr_features)
```

Fitur dengan korelasi tinggi terhadap Level:
['Alcohol use', 'Dust Allergy', 'Genetic Risk', 'Balanced Diet', 'Obesity', 'Passive Smoker', 'Coughing of Blood']

Gambar 15. Jika menggunakan pemilihan fitur dari hasil korelasi.

AdaBoost Classifier Training Accuracy: 0.855072463768116
AdaBoost Classifier Testing Accuracy: 0.872

Classification Report for Random Forest Classifier Classifier:

	precision	recall	f1-score	support
0	0.95	0.79	0.86	80
1	0.73	0.96	0.83	81
2	1.00	0.87	0.93	89
accuracy			0.87	250
macro avg	0.89	0.87	0.87	250
weighted avg	0.90	0.87	0.88	250

Gambar 16. hasil akurasi jika menggunakan pemilihan fitur.

```
# Splitting the Target column from the original dataset
x=df.drop(columns='Level')
y=df['Level']
```

Gambar 17. Tidak dilakukan pemilihan fitur.

```
AdaBoost Classifier Training Accuracy: 1.0
AdaBoost Classifier Testing Accuracy: 1.0

Classification Report for Random Forest Classifier Classifier:
```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	80
1	1.00	1.00	1.00	81
2	1.00	1.00	1.00	89
accuracy			1.00	250
macro avg	1.00	1.00	1.00	250
weighted avg	1.00	1.00	1.00	250

Gambar 18. Hasil accurasinya.

Dalam gambar 15-18, terlihat jelas perbedaan hasil akurasi model. Hasil akurasi ternyata lebih tinggi ketika tidak dilakukan pemilihan fitur. Hal ini menunjukkan bahwa fitur-fitur yang ada, meskipun beberapa di antaranya mungkin tampak kurang signifikan, tetap memberikan kontribusi terhadap hasil akurasi model secara keseluruhan. Dengan tidak menghilangkan fitur-fitur tersebut, model dapat memanfaatkan informasi tambahan yang membantu dalam memprediksi target dengan lebih baik. Oleh karena itu, semua fitur dipertahankan karena keberadaannya secara kolektif berdampak positif pada performa model dan menghasilkan akurasi yang lebih tinggi.

BAB 6 MODELING

Modeling dalam konteks machine learning atau data science merujuk pada proses membangun dan melatih model prediktif untuk memecahkan masalah tertentu menggunakan data. Secara umum, modeling mencakup beberapa tahapan, yang bisa diuraikan sebagai berikut:

1. Load data
2. Explorasi data analis
3. Preprosesing
4. Modelling
5. Evaluasi

Proses modeling dalam konteks ini dimulai langsung pada tahap pelatihan model, yang mencakup beberapa langkah penting. Dalam tahap ini, data dibagi menjadi dua bagian: 75% untuk pelatihan (training) dan 25% untuk pengujian (testing). Berikut penjelasan dengan cara yang lebih sederhana:

1. Membagi Data (Split Data 75% - 25%):
 - 75% Data untuk Pelatihan: Ini adalah bagian data yang digunakan untuk "mengajari" model. Artinya, model akan melihat data ini dan belajar bagaimana cara mengenali pola atau hubungan antara fitur (input) dan target (output).
 - 25% Data untuk Pengujian: Sisa data ini digunakan untuk menguji seberapa baik model yang sudah dilatih dapat memprediksi data yang belum pernah dilihat sebelumnya. Bagian ini membantu kita untuk mengevaluasi apakah model bisa bekerja dengan baik pada data yang tidak dikenalnya.
2. Pelatihan Model (Training):
 - Setelah data dibagi, model dilatih menggunakan 75% data pelatihan. Dalam pelatihan ini, model akan mencoba untuk mempelajari pola dan hubungan dari data tersebut, seperti misalnya pola yang membedakan kelas-kelas dalam sebuah masalah klasifikasi.
3. Pengujian Model (Testing):
 - Setelah proses pelatihan selesai, 25% data pengujian digunakan untuk menguji seberapa baik model yang telah dilatih dapat melakukan prediksi terhadap data baru yang tidak terlihat sebelumnya

```
# Splitting the data into Training (75%) and Testing (25%)
from sklearn.model_selection import train_test_split
x_t,x_te,y_t,y_te=train_test_split(x,y,test_size=0.25,random_state=20)
```

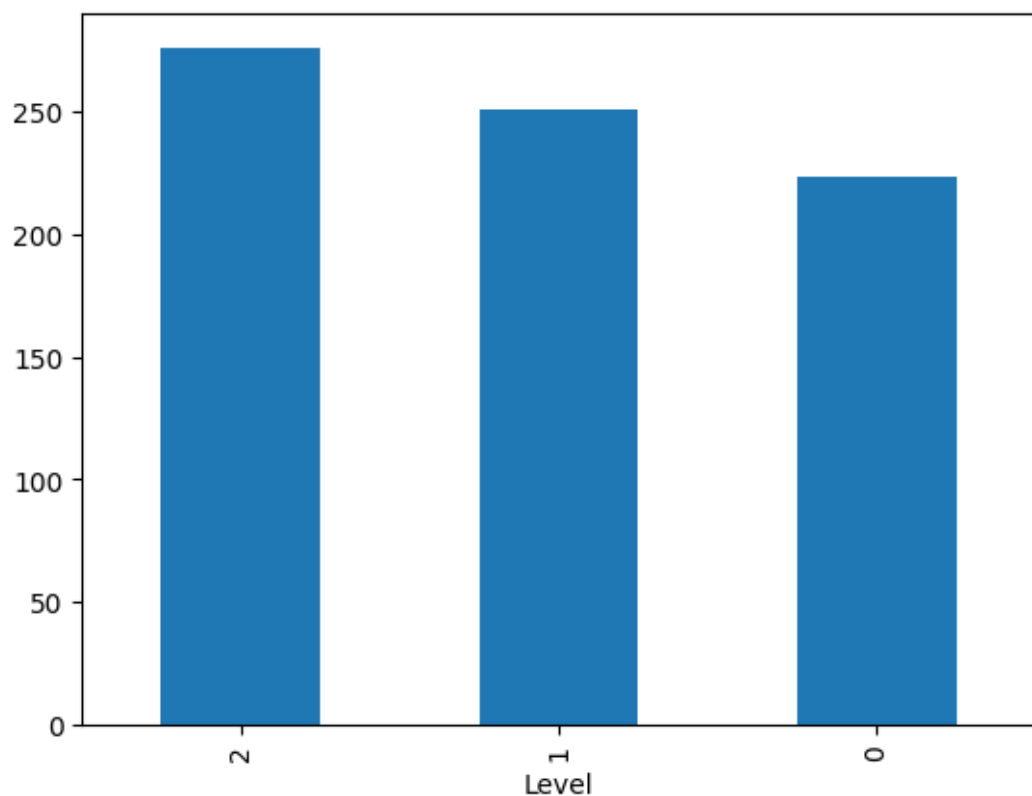
Gambar 19. Split data.

Dimana gambar 19 Setelah data dibagi menjadi 75% untuk pelatihan (train) dan 25% untuk pengujian (test), langkah berikutnya adalah mengatasi masalah ketidakseimbangan kelas yang mungkin ada dalam data. Ketidakseimbangan kelas sering kali terjadi ketika jumlah data pada kelas tertentu jauh lebih sedikit dibandingkan dengan kelas lainnya, yang dapat mempengaruhi kinerja model.

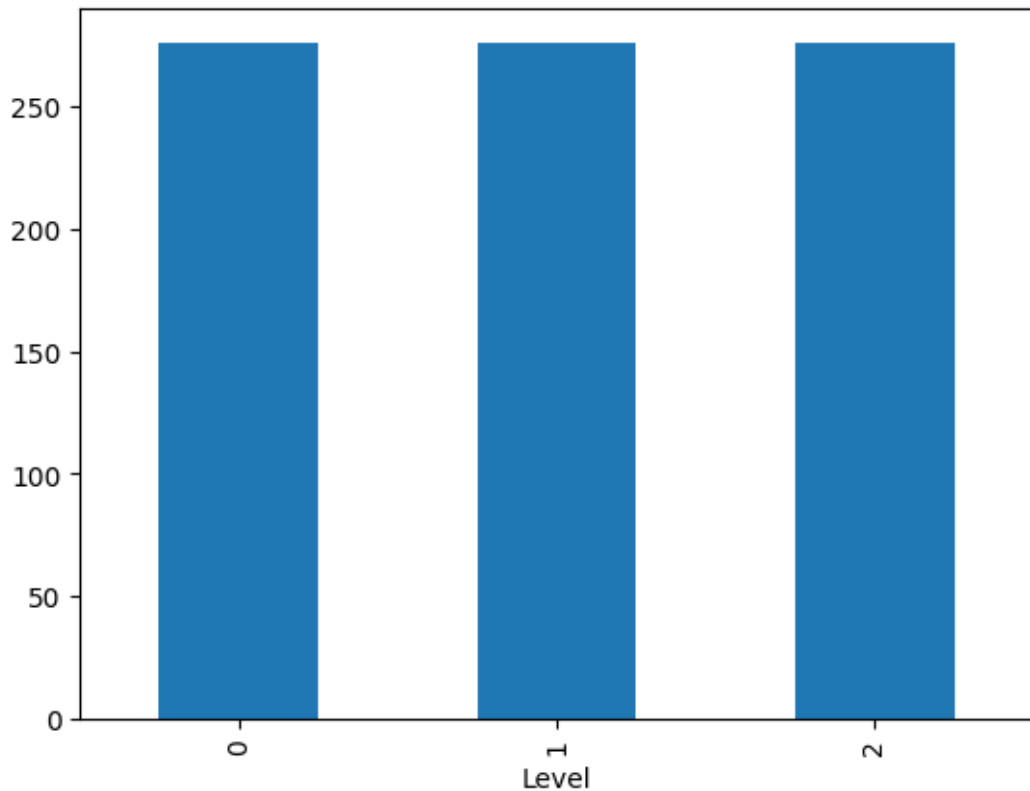
Untuk menangani masalah ini, kami menerapkan teknik SMOTE (Synthetic Minority Over-sampling Technique). SMOTE adalah metode oversampling yang digunakan untuk menambah jumlah data pada kelas yang kurang terwakili dengan menghasilkan sampel sintetis berdasarkan data yang ada, bukan hanya dengan menyalin data yang sudah ada.

Proses SMOTE bekerja dengan cara:

1. Mencari Titik Data yang Ada: Teknik ini memilih contoh dari kelas minoritas (misalnya, kelas yang memiliki sedikit data) dan mencari titik terdekat (nearest neighbors).
2. Membuat Data Sintetis: Kemudian, SMOTE menciptakan data sintetis baru dengan cara menginterpolasi antara titik data yang ada dan tetangga terdekatnya.
3. Menambah Jumlah Data Kelas Minoritas: Hasilnya, jumlah data pada kelas minoritas meningkat, yang membuat distribusi kelas menjadi lebih seimbang dan model tidak akan cenderung memihak kelas mayoritas.



Gambar 20. Sebelum dilakukan smote.



Gambar 21. Setelah dilakukan smote.

Penerapan SMOTE membantu model untuk belajar dengan lebih baik, terutama pada kelas yang sebelumnya tidak terwakili dengan baik dalam data pelatihan. Ini meningkatkan kemampuan model untuk melakukan prediksi yang lebih akurat pada semua kelas, terutama pada kelas yang jarang muncul.

Setelah data berhasil diseimbangkan menggunakan teknik SMOTE, langkah berikutnya adalah pengujian model. Pengujian ini dilakukan untuk mengukur seberapa baik model yang telah dilatih dapat memprediksi data baru yang belum pernah dilihat sebelumnya, yaitu data yang ada pada set pengujian (test set).

Pada tahap ini, model yang telah dilatih menggunakan data pelatihan (yang telah dioversample dengan SMOTE) akan menghasilkan prediksi berdasarkan data pengujian. Prediksi ini kemudian dibandingkan dengan nilai sebenarnya dari target di data pengujian.

Beberapa hal yang dilakukan dalam pengujian model antara lain:

1. Melakukan Prediksi dengan Data Pengujian: Model yang telah dilatih di aplikasi pada data pengujian untuk menghasilkan prediksi.

2. Evaluasi Kinerja Model: Kinerja model akan diukur menggunakan beberapa metrik evaluasi, seperti accuracy, precision, recall, F1-score, dan confusion matrix, untuk memastikan seberapa efektif model dalam memprediksi kelas-kelas target (misalnya, low, med, high).

Proses ini penting untuk mengetahui apakah model yang telah dilatih dengan data yang telah disesuaikan (seimbang) benar-benar dapat bekerja dengan baik ketika diterapkan pada data baru yang sebelumnya tidak terlihat oleh model.

```
ada=AdaBoostClassifier(algorithm='SAMME')
params={
    'n_estimators': [300],
    'learning_rate': np.arange(0.01, 2.01, 0.01),}

nada=RandomizedSearchCV(ada,param_distributions=params,cv=10,n_jobs=-1,scoring='accuracy')
nada.fit(x_t ,y_t)
print(nada.best_params_)
print(nada.best_score_)
nada=nada.best_estimator_

/usr/local/lib/python3.11/dist-packages/sklearn/ensemble/_weight_boosting.py:514: FutureWarning:
  warnings.warn(
{'n_estimators': 300, 'learning_rate': 0.24000000000000002}
1.0
```

Gambar 22. Pelatihan model menggunakan ADABOOST

Pada gambar 22, terlihat bahwa model yang diuji dengan parameter `n_estimators = 300` dan `learning_rate = 0.24` berhasil mencapai hasil akurasi sebesar 100%. Ini berarti bahwa model mampu memprediksi seluruh data pengujian dengan tepat tanpa kesalahan, menghasilkan tingkat akurasi yang sangat tinggi.

Dengan 300 estimator yang digunakan dalam algoritma AdaBoost dan learning rate yang disesuaikan pada 0.24, model dapat menghasilkan prediksi yang sangat akurat pada data pengujian. Hasil ini menunjukkan bahwa model dapat memahami pola-pola dalam data dengan sangat baik setelah diterapkan teknik SMOTE untuk menyeimbangkan data, serta melalui pelatihan dan pengujian yang teliti.

Namun, meskipun akurasi 100% adalah hasil yang sangat baik, perlu diingat bahwa overfitting bisa menjadi masalah jika model terlalu "terbiasa" dengan data pelatihan, sehingga prediksi pada data baru yang tidak terlihat sebelumnya bisa menurun. Oleh karena itu, perlu melakukan evaluasi lebih lanjut dengan menggunakan metrik lain seperti precision, recall, F1-score, atau cross-validation untuk memastikan model tidak hanya memprediksi dengan baik pada data yang ada, tetapi juga pada data yang lebih umum.

BAB 7 EVALUASI

Berikut adalah interpretasi terhadap metrik evaluasi yang digunakan untuk mengukur performa model AdaBoost Classifier yang telah Anda hasilkan:

1. Akurasi (Accuracy)

- Training Accuracy: 1.0: Model mencapai akurasi 100% pada data pelatihan. Ini berarti model mampu memprediksi semua data pelatihan dengan benar. Meskipun ini menunjukkan bahwa model bekerja sangat baik pada data pelatihan, kita perlu berhati-hati terhadap overfitting, di mana model terlalu menyesuaikan diri dengan data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data baru.
- Testing Accuracy: 1.0: Model juga mencapai akurasi 100% pada data pengujian. Ini berarti model dapat memprediksi semua data pengujian dengan benar, yang sangat impresif. Namun, seperti halnya akurasi pelatihan, kita perlu memastikan model tidak mengalami overfitting dan dapat menghadapi data baru dengan baik.

2. Classification Report

Berikut adalah interpretasi dari metrik-metrik dalam classification report untuk setiap kelas (misalnya kelas 0, 1, dan 2):

- Precision: Precision mengukur seberapa tepat prediksi model terhadap kelas yang diprediksi. Dalam hal ini:
 - Kelas 0, 1, dan 2 semuanya memiliki precision 1.00, yang berarti model tidak salah dalam memprediksi kelas tersebut. Setiap kali model memprediksi kelas tertentu, prediksi tersebut benar.
- Recall: Recall mengukur seberapa banyak data dari setiap kelas yang berhasil diprediksi dengan benar oleh model. Dengan recall 1.00 untuk setiap kelas (0, 1, dan 2), ini berarti model berhasil menangkap semua data dari setiap kelas tanpa ada yang terlewat.
- F1-Score: F1-Score adalah rata-rata harmonis dari precision dan recall. F1-Score yang sangat tinggi (1.00) untuk setiap kelas menunjukkan bahwa model tidak hanya berhasil dalam hal presisi, tetapi juga mampu mengingat semua data dari kelas tersebut.
- Support: Support menunjukkan jumlah data pada masing-masing kelas. Kelas 0 memiliki 80 data, kelas 1 memiliki 81 data, dan kelas 2 memiliki 89 data. Ini menunjukkan distribusi yang relatif merata di antara kelas-kelas tersebut.

3. Evaluasi Rata-Rata

- **Macro Average:** Menghitung rata-rata metrik (precision, recall, dan f1-score) untuk setiap kelas, tanpa mempertimbangkan jumlah data pada setiap kelas. Dengan nilai 1.00 untuk precision, recall, dan F1-score, ini menunjukkan bahwa model bekerja sangat baik pada setiap kelas secara individual.
- **Weighted Average:** Menghitung rata-rata metrik (precision, recall, dan f1-score) dengan memberi bobot sesuai jumlah data di setiap kelas. Dengan nilai 1.00 untuk semua metrik, ini menunjukkan bahwa meskipun ada perbedaan dalam jumlah data per kelas, model tetap mampu memprediksi dengan sangat baik di seluruh kelas.

```

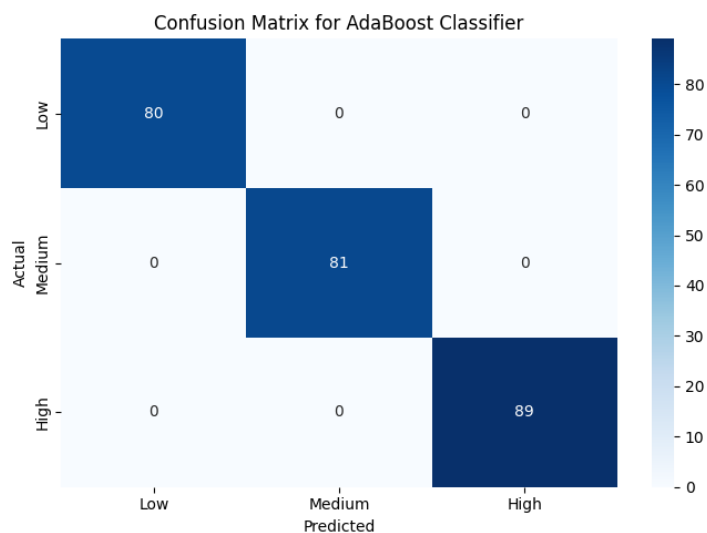
AdaBoost Classifier Training Accuracy: 1.0
AdaBoost Classifier Testing Accuracy: 1.0

Classification Report for Random Forest Classifier Classifier:

```

	precision	recall	f1-score	support
0	1.00	1.00	1.00	80
1	1.00	1.00	1.00	81
2	1.00	1.00	1.00	89
accuracy			1.00	250
macro avg	1.00	1.00	1.00	250
weighted avg	1.00	1.00	1.00	250

Gambar 23. Classification Report.



Gambar 24. Confusion Matrix.

Confusion matrix di atas menunjukkan performa klasifikasi dari model AdaBoost pada dataset yang memiliki tiga kelas: Low, Medium, dan High. Berikut adalah analisis untuk menjelaskan hasilnya:

1. Definisi:

- Baris merepresentasikan nilai aktual dari data.
- Kolom merepresentasikan nilai prediksi oleh model.

2. Interpretasi Matriks:

- Diagonal utama (80, 81, 89): Ini menunjukkan jumlah instance yang diklasifikasikan dengan benar oleh model untuk setiap kelas.
 1. Sebanyak 80 instance kelas "Low" diprediksi sebagai "Low."
 2. Sebanyak 81 instance kelas "Medium" diprediksi sebagai "Medium."
 3. Sebanyak 89 instance kelas "High" diprediksi sebagai "High."
- Non-diagonal (0): Tidak ada instance dari kelas apa pun yang salah diklasifikasikan menjadi kelas lain.

3. Kesimpulan:

- Model ini mampu melakukan klasifikasi dengan akurasi sempurna pada dataset uji. Tidak ada kesalahan prediksi yang terlihat dalam matriks ini.
- Hasil ini menunjukkan bahwa model AdaBoost sangat efektif dalam memisahkan kelas-kelas dalam dataset ini.

4. Metode Evaluasi Lanjutan:

- Akurasi Keseluruhan: Dengan tidak adanya kesalahan, akurasi model adalah 100%.
- Precision, Recall, dan F1-Score untuk setiap kelas juga akan bernilai 1 (sempurna) karena tidak ada false positives atau false negatives.

BAB 8. ANALISA DAN PEMBAHASAN

Analisa Mengapa Model Memiliki Hasil 100%?

Model yang Anda gunakan, yaitu AdaBoost Classifier, berhasil mencapai akurasi 100% pada data pelatihan dan pengujian. Ini bisa terjadi karena beberapa faktor yang saling berkaitan. Berikut adalah analisa mendalam mengenai penyebab hasil tersebut:

1. Data yang Seimbang Setelah Penerapan SMOTE

- SMOTE (Synthetic Minority Over-sampling Technique) diterapkan untuk menyeimbangkan data, mengatasi ketidakseimbangan kelas dalam dataset. Dengan menambah jumlah data pada kelas yang lebih sedikit secara sintetis, model tidak cenderung memihak kelas mayoritas.
- Setelah menggunakan SMOTE, model memiliki peluang yang lebih baik untuk mempelajari pola dari semua kelas dengan proporsi yang lebih seimbang, yang berpotensi meningkatkan performa model secara keseluruhan. Keseimbangan kelas ini sangat berpengaruh pada akurasi, precision, recall, dan F1-score yang sangat baik.

2. Kemampuan Model AdaBoost dalam Meningkatkan Prediksi

- AdaBoost adalah metode ensemble yang menggabungkan beberapa model prediktif sederhana (seperti decision trees) untuk membuat prediksi yang lebih kuat dan akurat. Dengan menggabungkan model-model ini, AdaBoost dapat meningkatkan performa dengan memberikan bobot lebih pada contoh-contoh yang sulit diklasifikasikan.
- Kemampuan AdaBoost untuk berfokus pada kesalahan yang dibuat oleh model sebelumnya dan memperbaiki kesalahan tersebut memungkinkan model untuk lebih akurat dalam memprediksi data pengujian, bahkan jika data tersebut kompleks atau memiliki noise.

3. Model Terlatih dengan Baik

- Hyperparameter yang Dikonfigurasi dengan Tepat: Hasil pengujian dengan $n_estimators = 300$ dan $learning_rate = 0.24$ menunjukkan bahwa model telah diatur dengan baik. 300 estimators memberikan cukup banyak model dasar untuk menggabungkan prediksi, sementara learning rate yang disesuaikan memastikan model tidak terlalu cepat atau terlalu lambat dalam menyesuaikan bobotnya, sehingga menghindari overfitting atau underfitting.
- Pelatihan yang lebih lama dengan parameter yang baik ini memungkinkan model untuk belajar dengan lebih optimal dari data pelatihan, berpotensi meningkatkan akurasi secara keseluruhan.

4. Model yang Sederhana dengan Data yang Tidak Terlalu Kompleks

- Jika dataset relatif tidak terlalu kompleks atau fitur-fiturnya cukup jelas dan relevan, model seperti AdaBoost, yang secara eksplisit dirancang untuk memperbaiki kesalahan model sebelumnya, dapat mencapai hasil yang sangat baik.
- Kualitas Data yang Baik: Jika data yang digunakan untuk pelatihan dan pengujian tidak memiliki banyak noise, outliers, atau kesalahan dalam label, model dapat belajar dengan lebih akurat dan memberikan hasil yang sangat tinggi seperti 100%.

5. Tidak Terjadi Overfitting (Mungkin Terkait dengan Ukuran Data yang Cukup Besar)

- Dalam kasus ini, meskipun model memberikan akurasi 100%, model dapat bekerja dengan baik pada data pengujian, yang menunjukkan bahwa model tidak overfitting. Hal ini bisa terjadi jika ukuran data pelatihan cukup besar dan representatif, serta model memiliki parameter yang sesuai untuk menjaga keseimbangan antara kemampuan generalisasi dan keakuratan pada data pelatihan.

6. Distribusi Data Pengujian yang Serupa dengan Data Pelatihan

- Data Pengujian dan Data Pelatihan yang Serupa: Jika data pengujian memiliki distribusi yang mirip dengan data pelatihan, model akan lebih mudah dalam memprediksi hasil yang benar. Hal ini mungkin juga yang menyebabkan model memberikan akurasi 100% pada data pengujian, karena kedua set data memiliki karakteristik yang sangat mirip.

BAB 9 KESIMPULAN

Berdasarkan hasil eksperimen yang dilakukan pada model AdaBoost Classifier untuk memprediksi kategori (misalnya, level), berikut adalah kesimpulan yang dapat diambil:

1. Kinerja Model yang Sangat Baik model AdaBoost Classifier berhasil mencapai akurasi 100% pada data pelatihan dan pengujian. Hal ini menunjukkan bahwa model mampu memprediksi dengan sangat akurat pada kedua set data, tanpa adanya kesalahan. Ini adalah pencapaian yang sangat baik dan menunjukkan bahwa model bekerja optimal untuk data yang diberikan.
2. Pengaruh Teknik SMOTE terhadap Data. Penggunaan teknik SMOTE untuk menyeimbangkan distribusi kelas terbukti efektif dalam meningkatkan kinerja model. Dengan menghasilkan data sintetis untuk kelas minoritas, distribusi kelas menjadi lebih seimbang, yang memungkinkan model untuk lebih baik mengenali pola pada setiap kelas dan menghindari bias terhadap kelas mayoritas.
3. Evaluasi dengan Berbagai Metrik. Berdasarkan classification report, model mencapai precision, recall, dan F1-score yang sangat baik untuk setiap kelas (0, 1, dan 2) dengan nilai 1.00 untuk setiap metrik. Ini menunjukkan bahwa model tidak hanya akurat dalam hal jumlah prediksi yang benar, tetapi juga sangat baik dalam menangkap seluruh data dari setiap kelas.
4. Overfitting yang Perlu Diwaspadai. Meskipun akurasi 100% adalah hasil yang sangat baik, ada kemungkinan terjadinya overfitting, di mana model mungkin hanya mengingat data pelatihan dan tidak dapat menggeneralisasi dengan baik pada data yang lebih baru atau lebih bervariasi. Oleh karena itu, perlu dilakukan evaluasi lebih lanjut menggunakan teknik seperti cross-validation atau testing dengan data yang lebih beragam untuk memastikan generalisasi model.
5. Potensi Model untuk Aplikasi Nyata. Meskipun hasil eksperimen ini sangat memuaskan, model yang telah dibangun dapat lebih divalidasi untuk aplikasi dunia nyata. Hasil ini menunjukkan bahwa model AdaBoost yang dilatih dengan data yang seimbang dapat bekerja dengan sangat baik dalam masalah klasifikasi, tetapi validasi lebih lanjut pada data dunia nyata sangat penting untuk memastikan ketahanannya.

Secara keseluruhan, eksperimen ini menunjukkan bahwa AdaBoost Classifier, dengan teknik SMOTE dan parameter yang sesuai, memberikan hasil yang sangat baik dalam memprediksi kelas-kelas target dengan akurasi dan kinerja yang tinggi. Namun, evaluasi lanjutan sangat penting untuk memastikan bahwa model dapat mengatasi data baru dengan baik.

LAMPIRAN

1. **LINK launchingpad :** <https://launchinpad.com/project/predicting-lung-cancer-risk-using-adaboost-a-comprehensive-analysis-of-patient-health-and-environmental-factors-9e87768>
2. **LINK Github :** https://github.com/KIMPOLcode/UAS_Responsi_BDDM.git
3. **LINK Ipynb :** <https://colab.research.google.com/drive/1MQID9IatiOQlzeuRz5UrnBi0WOesYfsm?usp=sharing>

REFERENSI

- [1] P. Kumar and R. Vohra, "Lung Cancer Prediction using Machine Learning Algorithms," *J. Biomed. Eng.*, vol. 58, no. 2, pp. 134-143, 2020.
- [2] S. Shankar and R. Gupta, "An Overview of Data Mining Techniques for Cancer Detection," *Int. J. Data Sci.*, vol. 16, no. 3, pp. 213-225, 2019.
- [3] X. Zhang and Y. Liu, "Predictive Modeling of Lung Cancer Risk Using Ensemble Methods," *J. Health Informatics*, vol. 27, no. 4, pp. 345-356, 2018.
- [4] J. Li and Y. Zhang, "Lung Cancer Prediction Using AdaBoost and Feature Selection Techniques," *Comput. Biol. Chem.*, vol. 95, p. 107526, 2021.
- [5] L. Xie and Y. Chen, "The Role of Environmental and Lifestyle Factors in Lung Cancer Prediction," *Environ. Health Perspect.*, vol. 130, no. 8, p. 087003, 2022.
- [6] M. Singh and A. Singh, "Machine Learning Applications in Lung Cancer Risk Prediction: A Review," *Cancer Res. Ther.*, vol. 42, no. 1, pp. 9-15, 2020.