

ANÁLISIS DE FACTORES RELACIONADOS CON EL CÁNCER DE PULMÓN

Desarrollo de un Modelo Predictivo mediante Machine
Learning

AUTORES:

valentina RODRÍGUEZ GÓMEZ, JESÚS DAVID JIMÉNEZ
ARANGO,
JOHANN ANDRÉS VELÁSQUEZ SÁNCHEZ, EVENCIO
VILLARRAGA Y SANTIAGO DÁVILA ARANGO


INSTITUCIÓN: MINTIC, UDEA - BOOTCAMP
INTELIGENCIA ARTIFICIAL

FECHA: 27 DE FEBRERO 2025




01

contexto



En Colombia, el cáncer de pulmón representa un problema de salud pública creciente. Según el Observatorio Global de Cáncer (Globocan), en 2022 se reportaron aproximadamente 7,000 nuevos casos de cáncer de pulmón en el país, con una tasa de mortalidad que supera los 6,000 fallecimientos anuales.

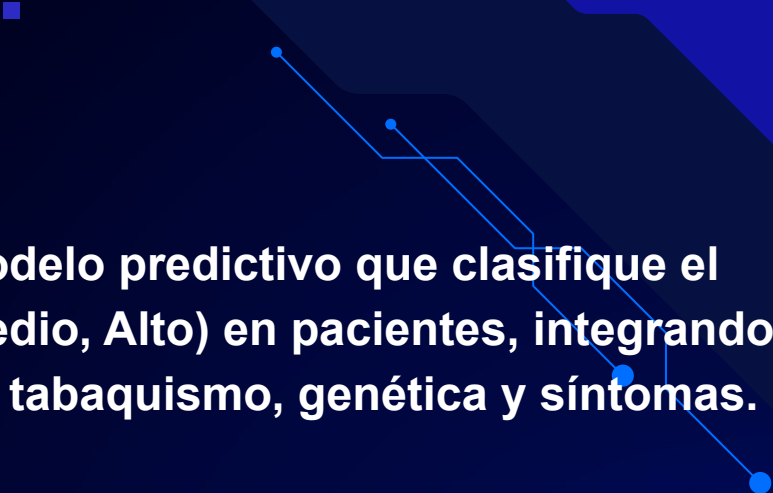

Esto lo convierte en uno de los tipos de cáncer más letales, con una tasa de supervivencia a cinco años inferior al 15%, cifra alarmante que refleja la necesidad de mejorar el acceso a diagnósticos tempranos y tratamientos oportunos.






02


OBJETIVO



Este proyecto busca desarrollar un modelo predictivo que clasifique el estadio de cáncer de pulmón (Bajo, Medio, Alto) en pacientes, integrando factores como contaminación del aire, tabaquismo, genética y síntomas.



Analizamos un dataset de 1.000 instancias para identificar patrones y crear una herramienta accesible que permita a los médicos actuar de manera preventiva y personalizada.



Análisis exploratorio de datos (EDA)

```
1 #Verificación de la información que contiene el dataset, valores nulos y/o repetidos.
2 dataCancer.info()
3 print('\nTotal de valores nulos: \n' + str(dataCancer.isnull().sum()))
4 print('\nTotal de valores duplicados: ' + str(dataCancer.duplicated().sum()))
```

```
<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   OccuPatinal 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
```

```
Total de valores nulos:
index                0
Patient Id           0
Age                  0
Gender               0
Air Pollution        0
Alcohol use          0
Dust Allergy         0
OccuPatinal Hazards  0
Genetic Risk         0
chronic Lung Disease 0
Balanced Diet        0
Obesity              0
Smoking              0
Passive Smoker       0
Chest Pain           0
Coughing of Blood    0
Fatigue              0
Weight Loss          0
Shortness of Breath  0
Wheezing             0
Swallowing Difficulty 0
Clubbing of Finger Nails 0
Frequent Cold        0
Dry Cough            0
Snoring              0
Level                0
dtype: int64
```

```
Total de valores duplicados: 0
```

```
[ ] 1 #Eliminación de las columnas index y Patient Id.
    2 dataCancer = dataCancer.drop(columns = ['index', 'Patient Id']);
    3 dataCancer
```



	Age	Gender	Air Pollution	Alcohol use	Dust Allergy	OccuPational Hazards	Genetic Risk	chronic Lung Disease	Balanced Diet	Obesity	...	Fatigue	Weight Loss	Shortness of Breath	Wheezing	Swallowing Difficulty	Clubbing of Finger Nails	Frequent Cold	Dry Cough	Snoring	Level
0	33	1	2	4	5	4	3	2	2	4	...	3	4	2	2	3	1	2	3	4	Low
1	17	1	3	1	5	3	4	2	2	2	...	1	3	7	8	6	2	1	7	2	Medium
2	35	1	4	5	6	5	5	4	6	7	...	8	7	9	2	1	4	6	7	2	High
3	37	1	7	7	7	7	6	7	7	7	...	4	2	3	1	4	5	6	7	5	High
4	46	1	6	8	7	7	7	6	7	7	...	3	2	4	1	4	2	4	2	3	High
...
995	44	1	6	7	7	7	7	6	7	7	...	5	3	2	7	8	2	4	5	3	High
996	37	2	6	8	7	7	7	6	7	7	...	9	6	5	7	2	4	3	1	4	High
997	25	2	4	5	6	5	5	4	6	7	...	8	7	9	2	1	4	6	7	2	High
998	18	2	6	8	7	7	7	6	7	7	...	3	2	4	1	4	2	4	2	3	High
999	47	1	6	5	6	5	5	4	6	7	...	8	7	9	2	1	4	6	7	2	High

1000 rows x 24 columns

CODIFICACIÓN DE LA VARIABLE OBJETIVO

Etiquetas codificadas:

Low : 1, Medium : 2, High : 3

```
1 # Se codifican con el método .map()
2 dataCancerCopy = dataCancer.copy()
3 mapeoClases = {'Low':0, 'Medium':1, 'High':2}
4 dataCancerCopy['Codificación'] = dataCancerCopy['Level'].map(mapeoClases)
5 print(dataCancerCopy.loc[:,['Level','Codificación']].head(20))
6 dataCancerCopy = dataCancerCopy.drop(columns=['Level']).rename(columns={'Codificación':'Level'});
```

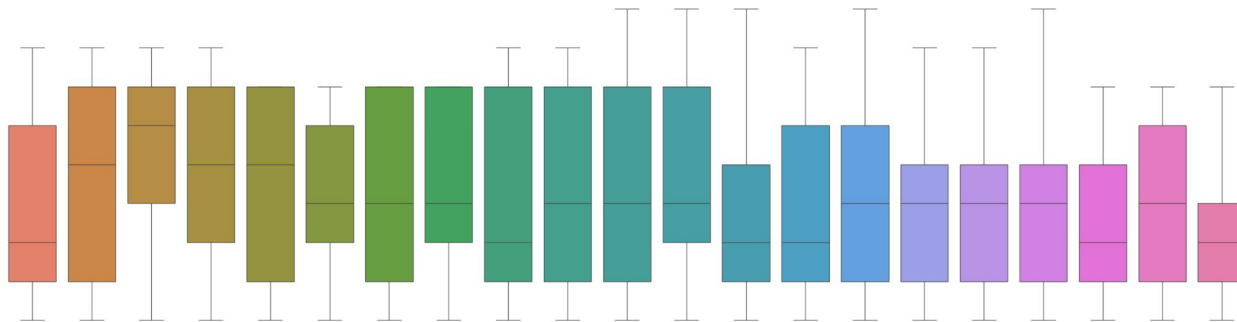
```
Level Codificación
0 Low 0
1 Medium 1
2 High 2
3 High 2
4 High 2
5 High 2
6 Low 0
7 Low 0
8 Medium 1
9 Medium 1
10 High 2
11 High 2
12 Medium 1
13 High 2
14 Low 0
15 Medium 1
16 Medium 1
17 High 2
18 High 2
19 Medium 1
```

```
1 dataCancerCopy.info()

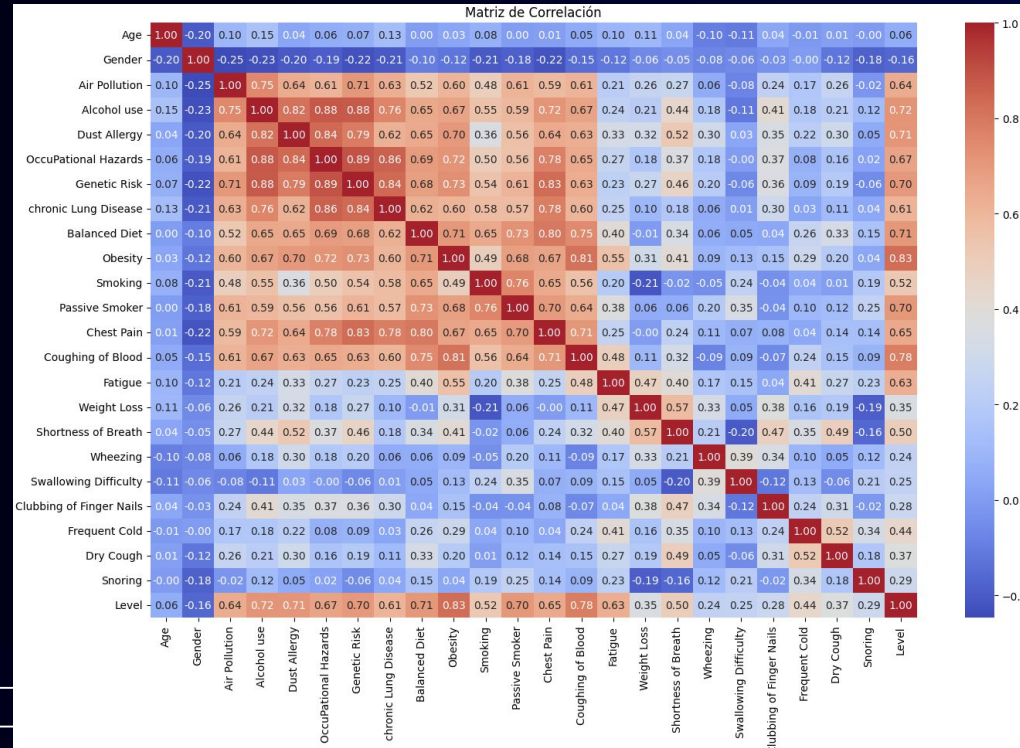
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 24 columns):
#   Column                                Non-Null Count  Dtype
---  ---                                -
0   Age                                  1000 non-null   int64
1   Gender                              1000 non-null   int64
2   Air Pollution                       1000 non-null   int64
3   Alcohol use                         1000 non-null   int64
4   Dust Allergy                       1000 non-null   int64
5   OccuPational Hazards               1000 non-null   int64
6   Genetic Risk                       1000 non-null   int64
7   chronic Lung Disease               1000 non-null   int64
8   Balanced Diet                      1000 non-null   int64
9   Obesity                            1000 non-null   int64
10  Smoking                            1000 non-null   int64
11  Passive Smoker                     1000 non-null   int64
12  Chest Pain                         1000 non-null   int64
13  Coughing of Blood                  1000 non-null   int64
14  Fatigue                           1000 non-null   int64
15  Weight Loss                        1000 non-null   int64
16  Shortness of Breath                1000 non-null   int64
17  Wheezing                           1000 non-null   int64
18  Swallowing Difficulty              1000 non-null   int64
19  Clubbing of Finger Nails           1000 non-null   int64
20  Frequent Cold                     1000 non-null   int64
21  Dry Cough                          1000 non-null   int64
22  Snoring                           1000 non-null   int64
23  Level                              1000 non-null   int64
dtypes: int64(24)
memory usage: 187.6 KB
```

ANÁLISIS DE VALORES ATÍPICOS Y/O SESGOS

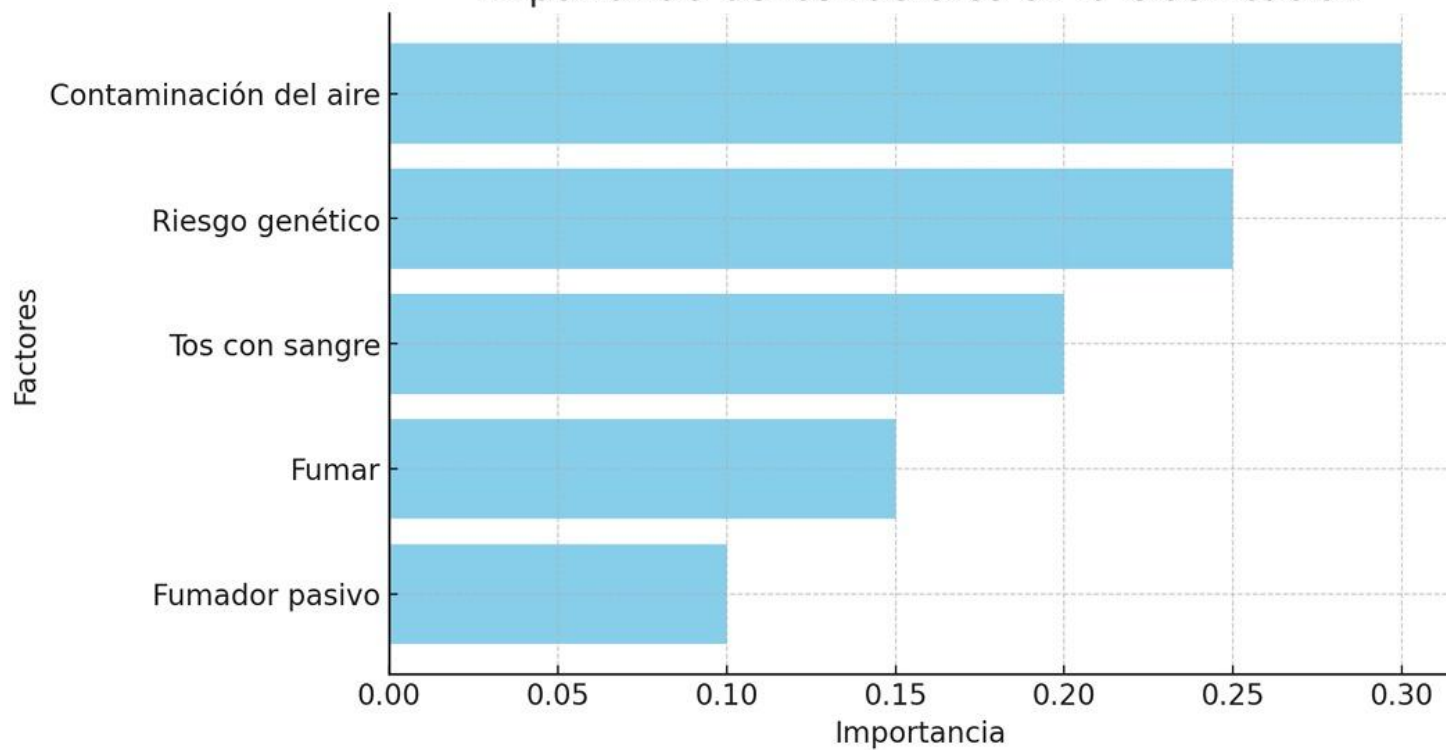
```
1 #Gráfica de bigotes para buscar valores atipicos y/o sesgos
2 plt.figure(figsize=(30,70))
3 sns.boxplot(data = dataCancerCopy.drop(columns = ['Gender','Level']))
4 plt.suptitle("Distribución de Características en el Conjunto de Datos de Cáncer", fontsize=40)
5 plt.xlabel('Características')
6 plt.ylabel('Valor')
7 plt.tight_layout()
8 plt.show()
```



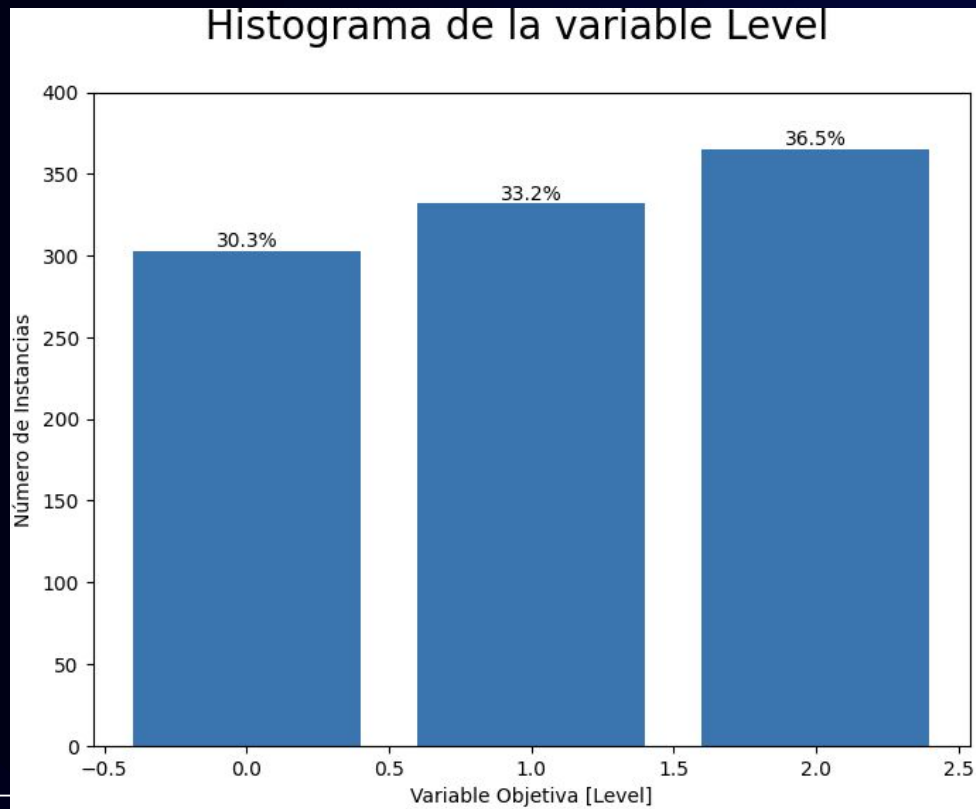
Análisis de correlación



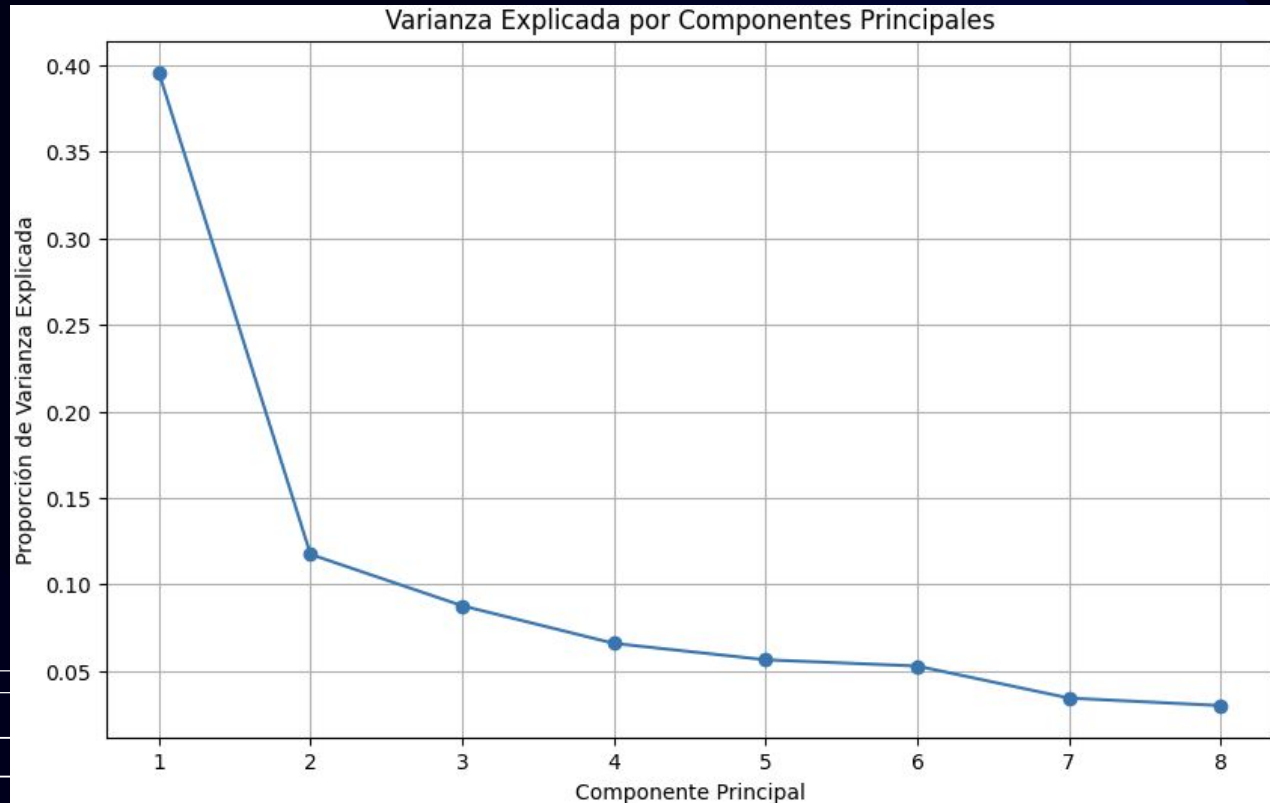
Importancia de los Factores en la Clasificación



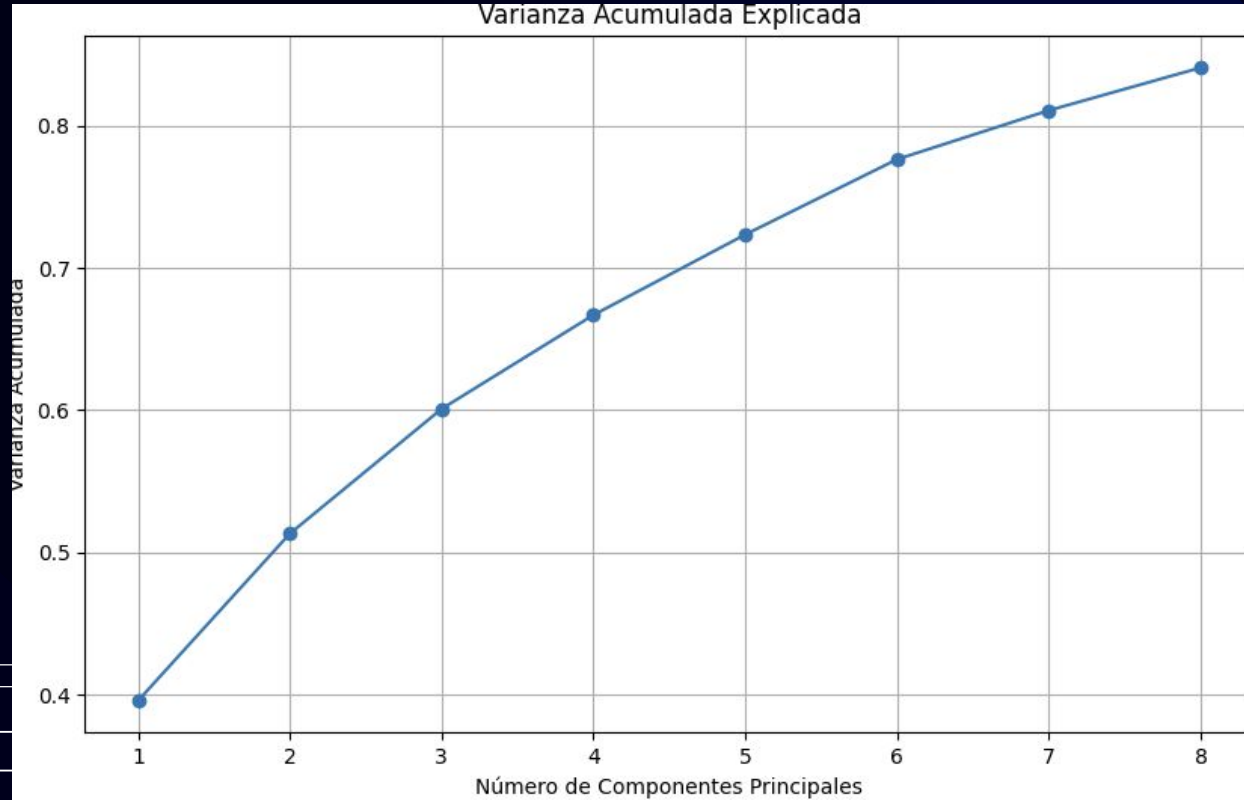
Balanceo de clases

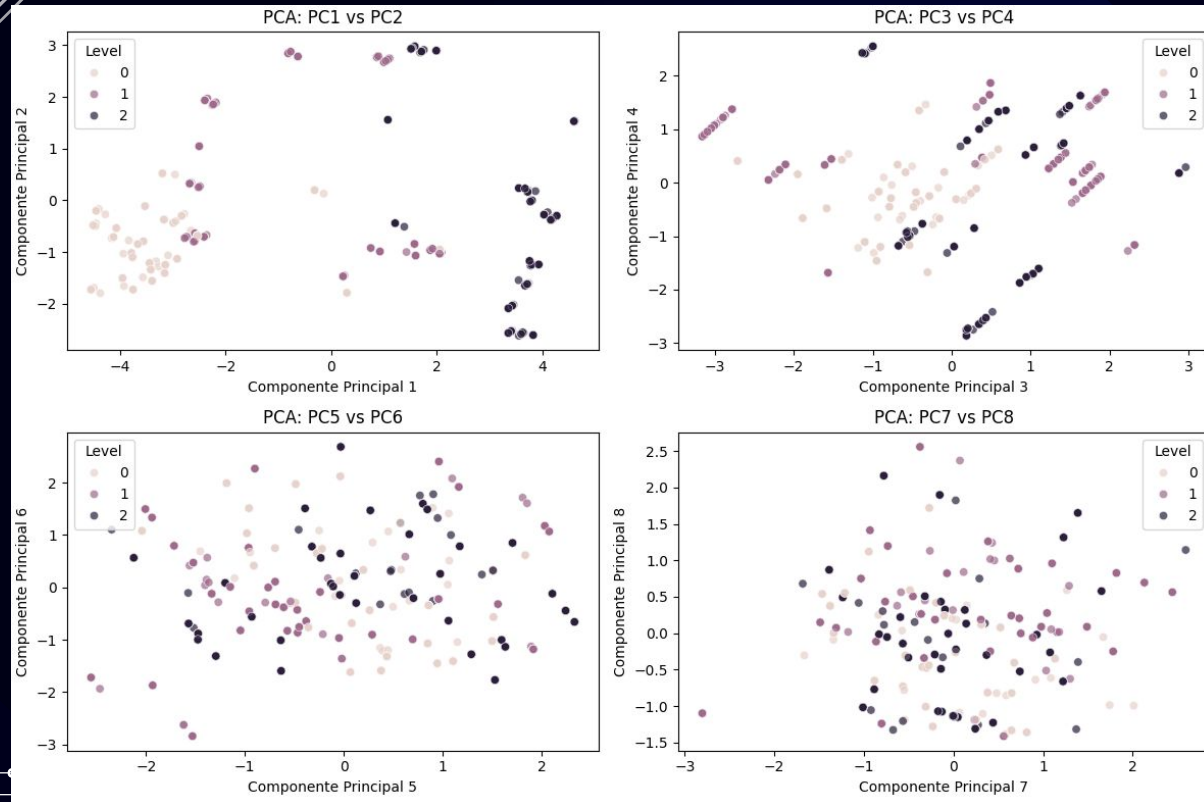


REDUCCIÓN DE DIMENSIONALIDAD CON PCA

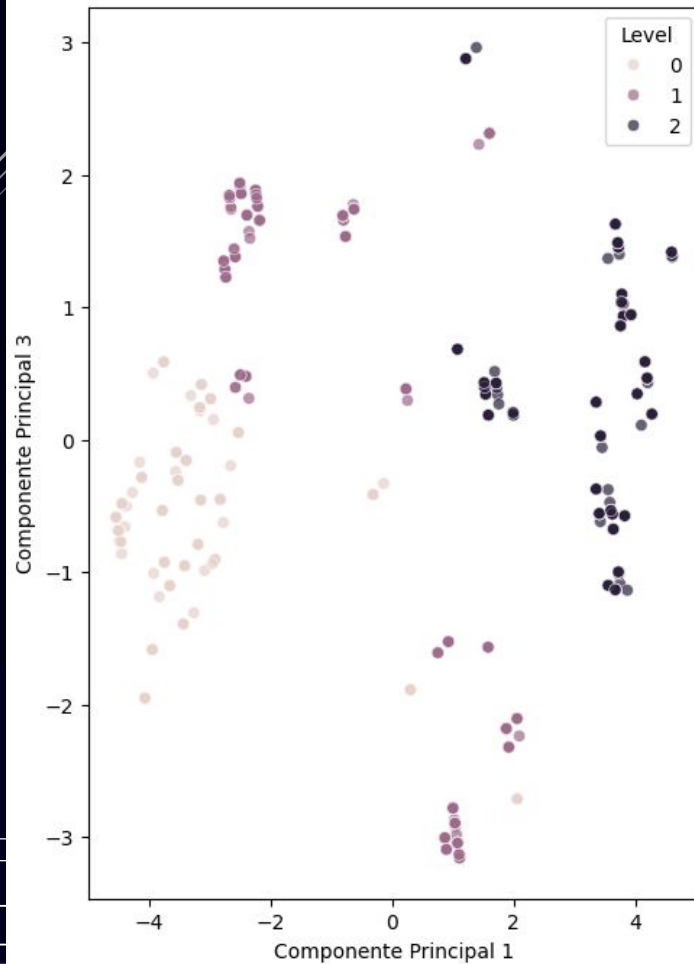


REDUCCIÓN DE DIMENSIONALIDAD CON PCA

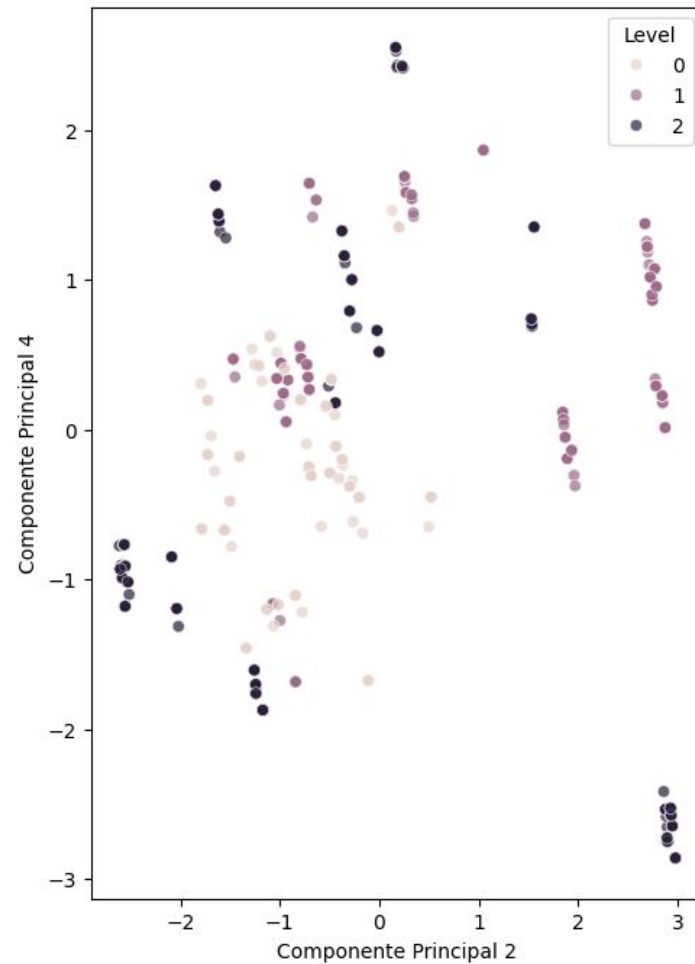




PCA: PC1 vs PC3



PCA: PC2 vs PC4



PCAS convertidos en DataFrame

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	Level
0	-2.532266	-0.685608	0.056141	-0.308613	-0.220362	0.737880	-0.945278	1.119602	0
1	-2.499235	1.046104	0.491503	1.867268	-0.048592	-0.965318	-2.804152	-1.099380	1
2	1.710896	2.877557	0.429408	-2.537105	1.172248	0.785246	-1.236978	0.491346	2
3	3.931440	-1.240071	0.945878	-1.762590	-1.569948	-0.688485	-0.140739	-0.490043	2
4	3.599374	-2.584447	-0.532676	-0.990434	-0.034350	0.647045	0.026444	-0.223923	2
...
995	3.713500	-1.619119	1.489598	1.439824	-0.233086	0.563613	-0.171922	-1.071613	2
996	4.032301	-0.276396	0.348624	1.001960	1.508345	0.324732	1.386575	1.651605	2
997	1.512550	2.928770	0.433276	-2.529480	2.331948	-0.658148	-0.320206	0.507402	2
998	3.355887	-2.564568	-0.370874	-0.768547	1.527850	-1.766072	0.140665	0.131219	2
999	1.990657	2.893786	0.205582	-2.728479	0.844396	1.490315	-0.831311	-0.015042	2

1000 rows x 9 columns



03

MODELO PRINCIPAL

Modelo Principal – Naive Bayes Gaussiano:

Se selecciona y crea el modelo GaussianNB().

Se entrena el modelo con el conjunto de entrenamiento (Modelo.fit(X_train, Y_train)).

Se realizan predicciones en el conjunto de prueba (Y_Predict = Modelo.predict(X_test)).

Evaluación:

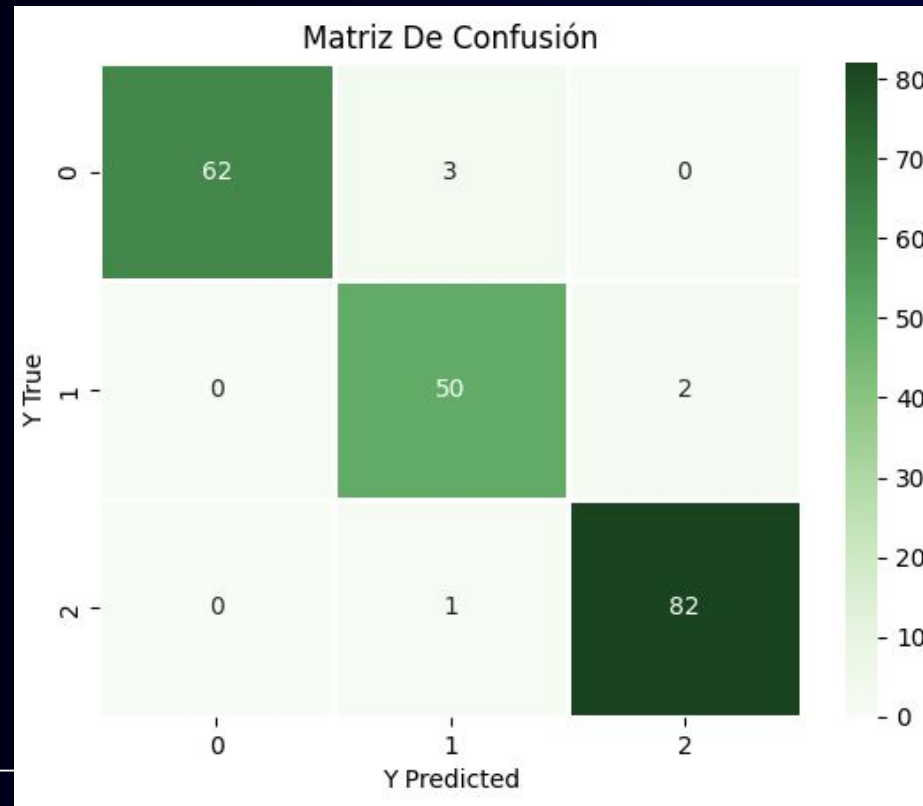
Se muestra la matriz de confusión mediante la función matrizConfusion().

Se calculan y presentan las métricas (accuracy, precision, recall y F1-score)

usando la función metrics(), mostrando un alto desempeño (por ejemplo, **~97%**

de exactitud).

EVALUACIÓN DEL MODELO



Conclusiones del modelo

Accuracy: El modelo predijo correctamente el 97.00% de las instancias en el conjunto de prueba.

■ Precisión: Todas las instancias que el modelo clasificó como pertenecientes a una clase específica, el 97.09% realmente pertenecían a esa clase. Esto indica que el modelo tiene una baja tasa de falsos positivos. ■

Recall: El modelo identificó correctamente el 97.00% de todas las instancias que realmente pertenecían a cada clase. Esto indica que el modelo tiene una baja tasa de falsos negativos.

F1-Score: Indica un buen equilibrio entre la precisión y el recall. Esto significa que el modelo es tanto preciso como sensible.



04

EXPerimentos adicionales

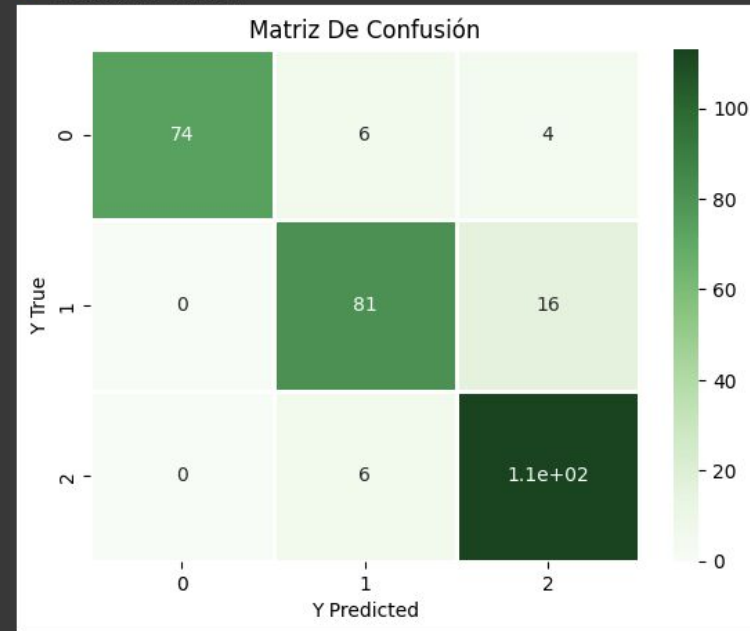
EXPERIMENTO #1 Naive Bayes DIVISION 70-30

random_state=42

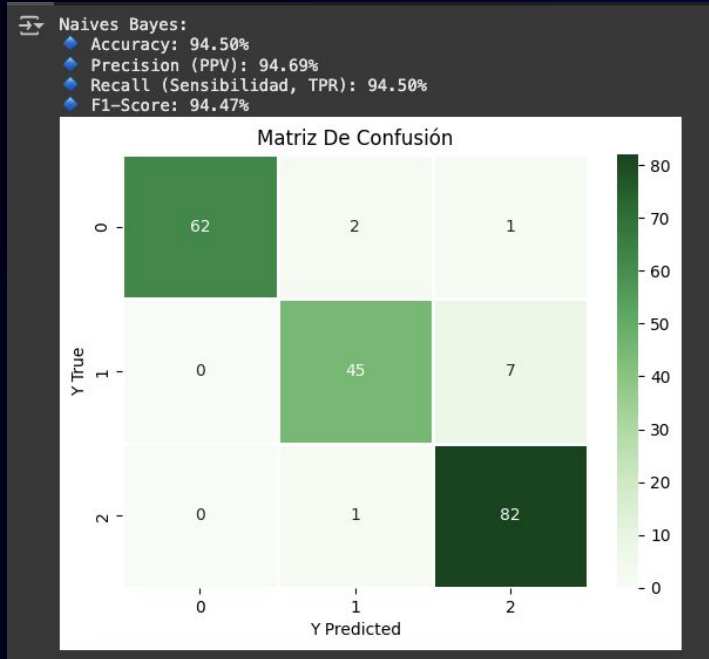


Naive Bayes:

- ◆ Accuracy: 89.33%
- ◆ Precision (PPV): 89.86%
- ◆ Recall (Sensibilidad, TPR): 89.33%
- ◆ F1-Score: 89.37%



EXPERIMENTO#2 Naive Bayes. se elimina la característica [occupational hazards] debido a COLINEALIDAD DIVISION 80-20 random_state=100



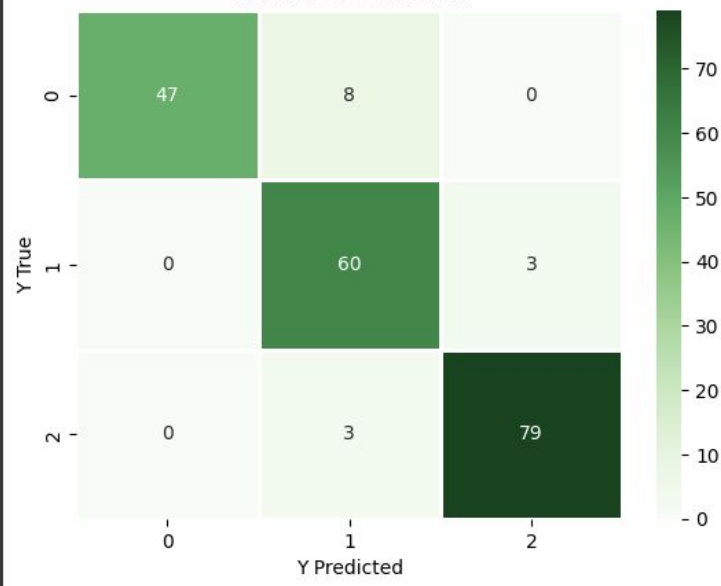
EXPERIMENTO #3 Naive Bayes con PCA con 8 componentes y valor de random_state = 42



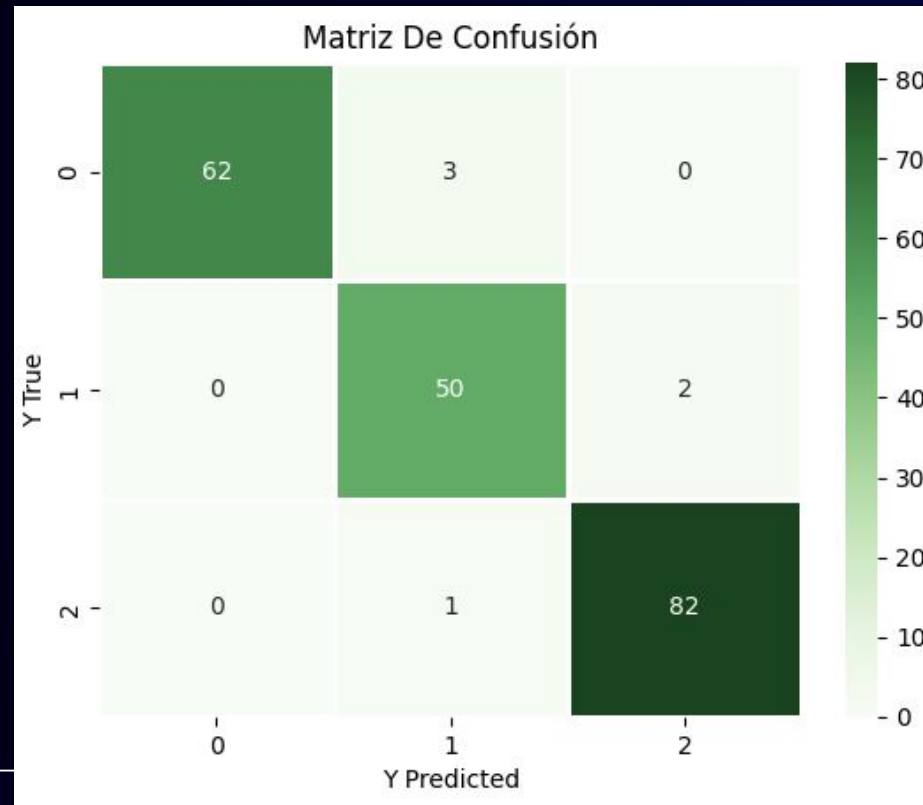
Resultados Naive Bayes:

- ◆ Accuracy: 93.00%
- ◆ Precision (PPV): 93.62%
- ◆ Recall (Sensibilidad, TPR): 93.00%
- ◆ F1-Score: 93.05%

Matriz De Confusión



EVALUACIÓN DEL MODELO



JUSTIFICACIÓN

Proporciona una metodología basada en datos para identificar y priorizar factores de riesgo de cáncer, lo que puede ser utilizado por profesionales de la salud para mejorar la prevención y el diagnóstico temprano.

Los modelos desarrollados tienen el potencial de ser integrados en sistemas de apoyo clínico, permitiendo una evaluación rápida y precisa del riesgo de cáncer en pacientes.

Contribuye a la literatura científica al establecer una relación cuantitativa entre la contaminación del aire, el tabaquismo, los factores genéticos y el riesgo de cáncer, lo que puede informar políticas públicas y estrategias de salud.



FIN

dataset:

[HTTPS://WWW.KAGGLE.COM/DATASETS/THE
DEVASTATOR/CANCER-PATIENTS-AND-AIR-P
OLLUTION-A-NEW-LINK/DATA](https://www.kaggle.com/datasets/the-devastator/cancer-patients-and-air-pollution-a-new-link/data)

GITHUB:

PICKLE: