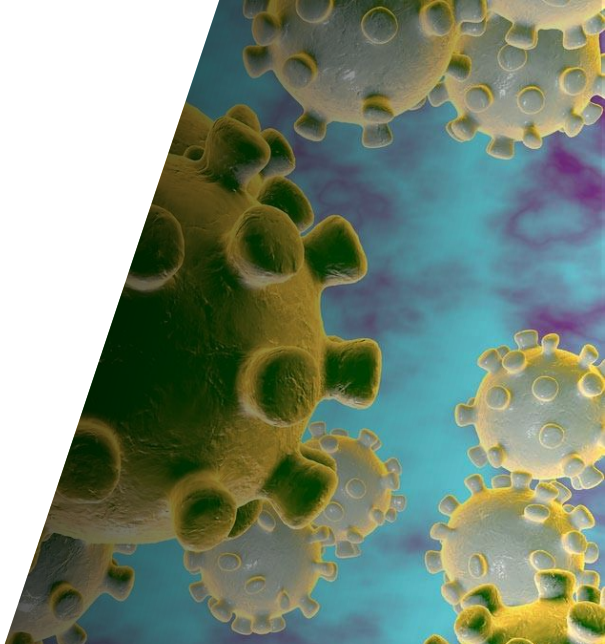


Proyecto: COVID-19

31508811-1 Martiñón Luna Jonathan José.
31506842-1 Ortega Ibarra Jaime Jesus.
41800471-9 Tapia López José de Jesús

Reconocimiento de patrones
10 de Junio del 2020



Objetivos y Ambiente

- Identificar pacientes COVID a través de radiografías y tomografías a través de modelos de clasificación.
- Comparar los daños presentes en los pulmones, comparando con tomografías de pulmones que no tengan COVID-19.
- A partir de las tomografías conocidas que tienen COVID-19, identificar las diversas regiones: Esmeralizado, consolidado y derrame pleural.
- Comparar rendimientos de modelos y estrategias para la clasificación y detección en imágenes.
- La práctica fue desarrollada en *Google Colab*, haciendo uso de Python 3.

Introducción

- COVID-19 es una enfermedad infecciosa causada por el virus SARS-CoV-2 y se detectó por primera vez en la ciudad china de Wuhan (provincia de Hubei) en diciembre de 2019.
- El COVID-19 ha llegado a casi todos los territorios, causando que el 11 de marzo de 2020 la Organización Mundial de la Salud la declaró pandemia.
- Organizaciones, universidades, centros de investigación, gobiernos han creado bases de datos abiertas que contienen datos de personas que hayan contraído la enfermedad.

Clasificador: KNN

También conocido como algoritmo de vecinos más cercanos. Es un algoritmo supervisado que nos ayudará a agrupar mediante ' K ' vecinos más próximos, es decir, podemos seleccionar la cantidad de elementos en un grupo. Ocupa bastante espacio de memoria, pero es ideal para pequeños datos.

Clasificador: KMeans

Dicho algoritmo es no supervisado; trabajará con ' K ' centroides, a diferencia del anterior, donde seleccionabamos la cantidad de elementos en nuestros grupos, aquí buscaremos tener ' K ' grupos. Se comienza con centroides aleatorios y se toma distancia con los puntos se decide que las menores correspondan a cierto grupo, finalmente de los puntos se calcula el promedio en (X,Y) y se obtienen nuevos centroides; termina en el momento en que se cumplan ciertas iteraciones, no haya cambio de centroides o hayamos conseguido cierto margen de error.

Clasificador: Naive Bayes

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)},$$

- *Bernoulli*: $\prod_k p_{i_k}^{x_k} (1 - p_{i_k})^{(1-x_k)}$.
- *Multinomial*: $\frac{(\sum_k x_k)!}{\prod_k x_k!} \prod_k p_{i_k}^{x_k}$, donde las muestras (vectores de características) representan las frecuencias con las que ciertos eventos han sido generados por un multinomio (p_1, \dots, p_n) , y p_i es la probabilidad de que ocurra el evento i .
- *Gaussiano*: $\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}}$.

Clasificador: SVM

Sirven para tratar con límites no lineales. Trabaja con *kernels*:

1. Lineal:

$$K(x_i, x'_i) = \sum_{j=1}^P x_{i,j}, x'_{i,j}$$

2. Polinómico:

$$K(x_i, x'_i) = \left(1 + \sum_{j=1}^P x_{i,j}, x'_{i,j} \right)^d$$

3. Radial:

$$K(x_i, x'_i) = \exp \left(-\gamma \sum_{j=1}^P (x_{i,j}, x'_{i,j})^2 \right)$$

Métricas

Después del resultado de una clasificación de registros, es posible asignarla a una de las siguientes 4 categorías:

- **Verdaderos positivos (VP):** Registros que han sido clasificados como *correspondientes* y que realmente sí lo son.
- **Falsos positivos (FP):** Registros que han sido clasificados como *correspondientes* pero NO lo son.
- **Verdaderos negativos (VN):** Registros que han sido clasificados como *no correspondientes*, y que en efecto no lo son.
- **Falsos negativos (FN):** Registros que han sido clasificados como *no correspondientes* y que de hecho sí son *correspondientes*.

Métrica: Precisión

¿Cuántos de los etiquetados como correspondientes sí son correspondientes?

$$\frac{VP}{VP + FP}$$

Métrica: Recall

¿Cuántos etiquetamos correctamente?

$$\frac{VP}{VP + FN}$$

Métrica: F1-Score

Debe existir una compensación entre Recall y Precisión. Por lo tanto, F1-score es el peso promedio de Presición y Recall:

$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

PRIMERA PARTE: CLASIFICACIÓN DEL TÓRAX

TORÁX PERSONA NORMAL VS TÓRAX PACIENTE DE COVID-19

Recopilación de Imágenes

Primero, recopilamos las imágenes de Rayos X de resultados positivos de paciente con coronavirus.

Kaggle contiene imágenes de rayos X de neumonía, COVID-19, pacientes normales, etc¹. Por lo tanto, para conseguir las imágenes de rayos X de coronavirus, nos fijamos en el CSV que contiene a los metadatos de las imágenes, ya que ahí se encuentra una descripción de cada imagen. Para ello descargamos dicho archivo y lo leemos:

¹Específicamente, están en el siguiente link:

Recopilación de Imágenes

	patientid	view	modality	folder	filename	url
finding						
ARDS	4	4	4	4	4	4
COVID-19	296	296	296	296	296	296
COVID-19, ARDS	12	12	12	12	12	12
Chlamydophila	2	2	2	2	2	2
E.Coli	4	4	4	4	4	4
Klebsiella	1	1	1	1	1	1
Legionella	2	2	2	2	2	2
No Finding	3	3	3	3	3	3
Pneumocystis	15	15	15	15	15	15
SARS	16	16	16	16	16	16
Streptococcus	17	17	17	17	17	17

Recopilación de Imágenes

Obtenemos las radiografías de los pacientes con COVID-19. Para eso, cambiaremos el conjunto de datos y seleccionamos las filas donde el resultado es igual a COVID-19, y la vista debe ser PA (Posterior Anterior).

En total, se tomaron 141 imágenes de rayos X de COVID-19 para los modelos que vamos a presentar.

Recopilación de Imágenes

Por otro lado, para tener imágenes de personas normales (sanas), sustraemos una muestra de 100^2 y las colocamos en una carpeta junto a la carpeta del conjunto de datos del COVID-19. Esto es, en una carpeta que creamos y nombramos como *Datos* se encuentran las carpetas *Normal* y *COVID_19*, las cuales contienen sus correspondientes imágenes.

²Obtenidas de: <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>

Indicando etiquetas

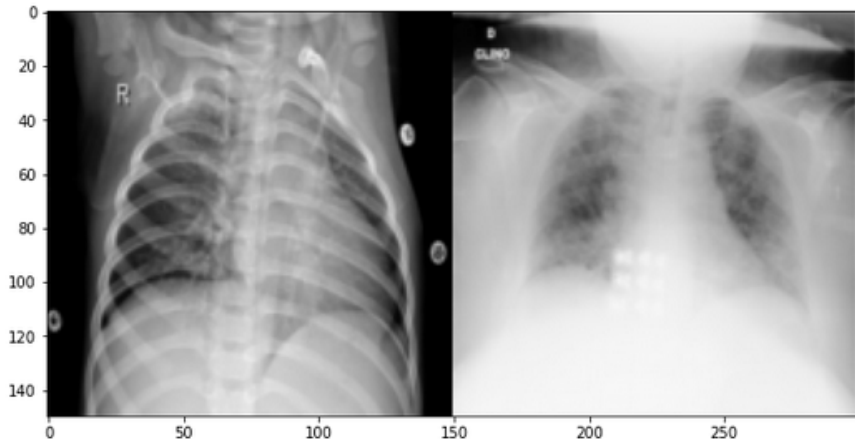
Procedimos a guardar cada imagen y su etiqueta: **COVID_19** o **Normal**, donde por obvias razones la primera indica si es sobre coronavirus y la segunda si es de un paciente normal. Las imágenes (ya como arreglos de NumPy) y las etiquetas las guardamos en dos listas separadas. Además, modificamos el tamaño de las imágenes, de tal forma que su dimensión fuera de 224x224, estuvieran en *RGB* y escalamos las intensidades de píxeles de cada imagen al rango [0,1].

Obtenemos el nombre de las imágenes

Ya que las hemos guardado con su correspondiente etiqueta para diferenciarlas, en seguida obtenemos el nombre de las imágenes de cada carpeta, por eso indicamos la ruta en la que se encuentran las dos carpetas (Una es de las imágenes de Rayos X del coronavirus y la otra es de Rayos X de personas normales).

Ejemplos de imágenes recopiladas

Radiografía de tórax Normal (sano) VS radiografía de tórax con COVID-19



Convertimos las etiquetas categóricas a numéricas

Es importante mencionar que las etiquetas de las imágenes están en forma de cadena, y para los algoritmos de aprendizaje de máquina se requiere que sean numéricas. Por esta razón, categorizamos cada etiqueta con un número:

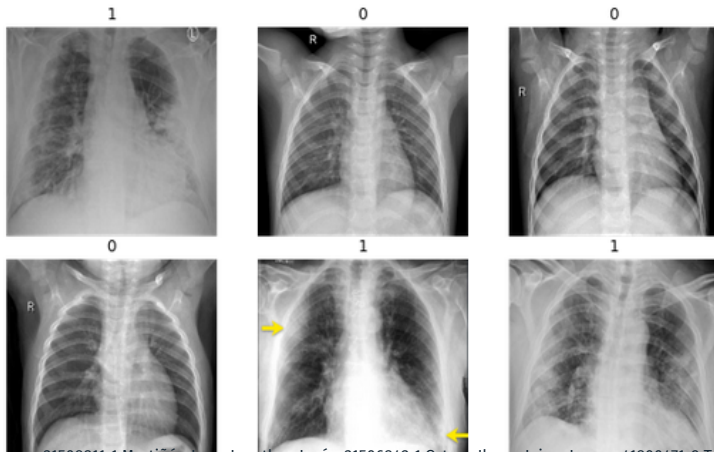
- 1, si la imagen es de COVID-19
- 0, si la imagen es de una persona sana.

Conjuntos de entrenamiento y prueba

Dividimos nuestros datos en conjuntos de entrenamiento y prueba, de tal manera que tenemos 180 imágenes de entrenamiento y 61 de prueba con las siguientes dimensiones para X_{train} , Y_{train} , X_{test} , Y_{test} : $((180, 224, 224, 3), (61, 224, 224, 3), (180,), (61,))$.

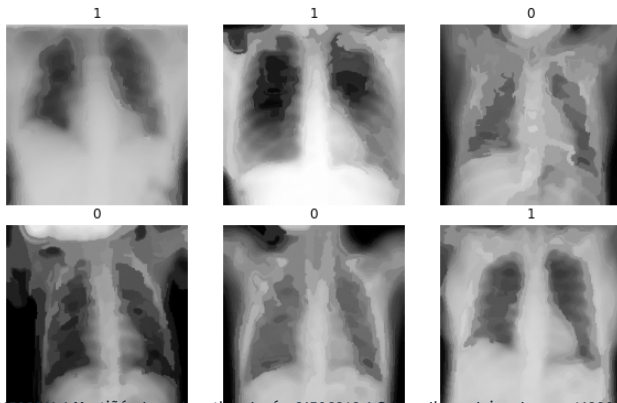
Para los algoritmos de clasificación, se debe tener en cuenta que los conjuntos que corresponden a la matriz de forma de las características deben tener una dimensión menor o igual a 2, por lo que haciendo un *reshape* a los conjuntos anteriores, obtenemos las siguientes dimensiones: $((180, 150528), (61, 150528), (180,), (61,))$.

Visualización de algunas figuras con su correspondiente etiqueta



Visualización de algunas figuras con filtro gaussiano y superpíxeles

Imágenes con filtro gaussiano y Superpíxeles:



KNN

```
# Creamos el clasificador KNN con 100 vecinos
knn = KNeighborsClassifier(n_neighbors = 100)
# Ajustamos el clasificador a los datos
Y_pred = knn.fit(x_train,Y_train).predict(x_test)
# Y evaluamos esta primera clasificación:
reporte = classification_report(Y_test, Y_pred)
print(reporte)
```

	precision	recall	f1-score	support
0	0.93	0.96	0.95	27
1	0.97	0.94	0.96	34
accuracy			0.95	61
macro avg	0.95	0.95	0.95	61
weighted avg	0.95	0.95	0.95	61

KNN: Interpretación de la evaluación

Clase 0 (Normal):

- *precision*: El 0.93 indica que el 93% de los registros de este clasificador han sido clasificados como "Normal" y que efectivamente corresponden a "Normal".
- *recall*: El 0.96 significa que se ha clasificado correctamente como "Normal" al 96% de los registros.

Clase 1 (COVID-19):

- *precision*: El 0.97 indica que el 97% de los registros de este clasificador han sido clasificados como "COVID-19" y que efectivamente corresponden a "COVID-19".
- *recall*: El 0.94 significa que se ha clasificado correctamente como "COVID-19" al 94% de los registros.

Naive Bayes: Multinomial

```
# Ajustamos el clasificador a los datos
# Creamos otro clasificador: Naive Bayes Multinomial
nbm = MultinomialNB()
Y_pred2 = nbm.fit(x_train,Y_train).predict(x_test)
# Generamos el reporte del modelo
reporte2 = classification_report(Y_test, Y_pred2)
print(reporte2)
```

	precision	recall	f1-score	support
0	0.93	0.96	0.95	27
1	0.97	0.94	0.96	34
accuracy			0.95	61
macro avg	0.95	0.95	0.95	61
weighted avg	0.95	0.95	0.95	61

Naive Bayes: Bernoulli

```
# Creamos otro clasificador Naive Bayes Bernoulli
nbb = BernoulliNB() Y_pred5 = nbb.fit(x_train,Y_train).predict(x_test)
# Generamos el reporte del modelo
reporte5 = classification_report(Y_test, Y_pred5)
print(reporte5)
```

	precision	recall	f1-score	support
0	0.91	0.78	0.84	27
1	0.84	0.94	0.89	34
accuracy			0.87	61
macro avg	0.88	0.86	0.86	61
weighted avg	0.87	0.87	0.87	61

Naive Bayes: Gaussiano

Creamos otro clasificador Naive Bayes Gauss

```
nbg = GaussianNB()
```

```
Y_pred3 = nbg.fit(x_train,Y_train).predict(x_test)
```

Generamos el reporte del modelo

```
reporte3 = classification_report(Y_test, Y_pred3)
```

```
print(reporte3)
```

	precision	recall	f1-score	support
0	0.96	0.96	0.96	27
1	0.97	0.97	0.97	34
accuracy			0.97	61
macro avg	0.97	0.97	0.97	61
weighted avg	0.97	0.97	0.97	61

SVM

```
# Creamos otro clasificador SVM svm = SVC()
Y_pred4 = svm.fit(x_train,Y_train).predict(x_test)
# Generamos el reporte
reporte4 = classification_report(Y_test, Y_pred4)
print(reporte4)
```

	precision	recall	f1-score	support
0	0.96	0.96	0.96	27
1	0.97	0.97	0.97	34
accuracy			0.97	61
macro avg	0.97	0.97	0.97	61
weighted avg	0.97	0.97	0.97	61

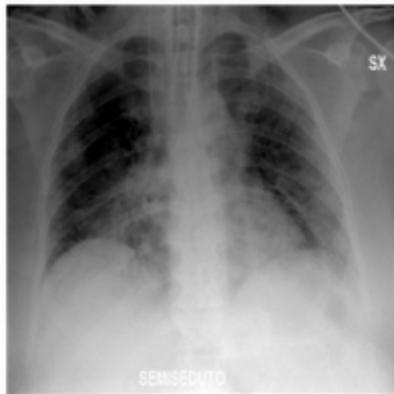
Predicciones

Para demostrar qué tan buenos fueron nuestros modelos, a continuación vamos a predecir con cada uno de estas dos imágenes: una de un paciente con coronavirus y otra de un paciente normal cabe mencionar que cada imagen, será procesada de dos maneras distintas, una será directo y otra mediante filtro gaussiano y superpíxeles. Realizamos dichas predicciones en el mismo orden en que presentamos los algoritmos: **KNN, Multinomial Naive Bayes, Bernoulli Naive Bayes, Gaussiano Naive Bayes y SVM.**

Iniciamos con la imagen de rayos X del tórax de una persona con covid 19:

Predicción: KNN

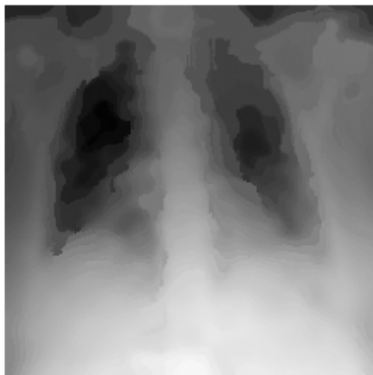
KNN
Predicción: COVID-19



Predicción: KNN

Imagen con Filtro gaussiano y superpíxeles

KNN
Predicción: COVID-19



Predicción: Naive Bayes Multinomial

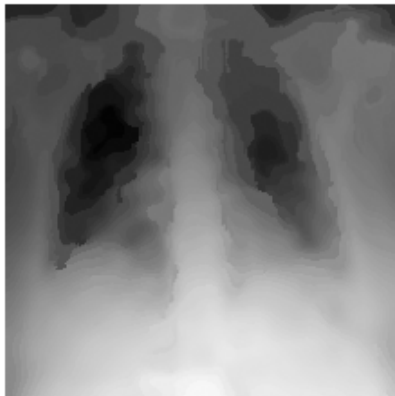
Multinomial Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Multinomial

Imagen con filtro gaussiano y superpíxeles

Multinomial Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Bernoulli

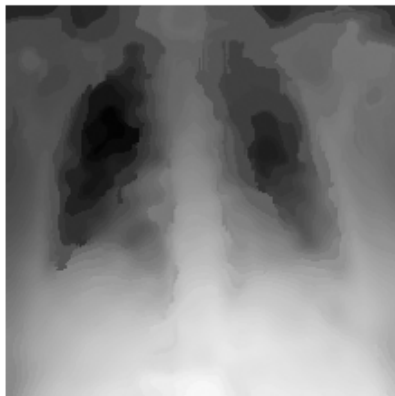
Bernoulli Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Bernoulli

Imagen con filtro gaussiano y superpíxeles

Bernoulli Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Gaussiano

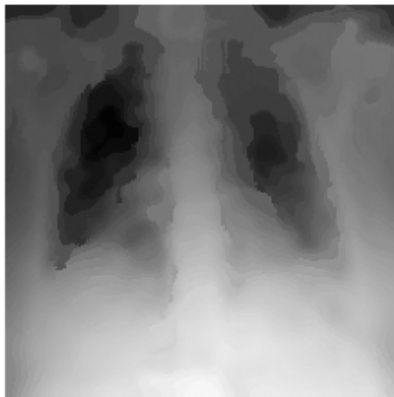
Gaussiano Naive Bayes
Predicción: COVID-19



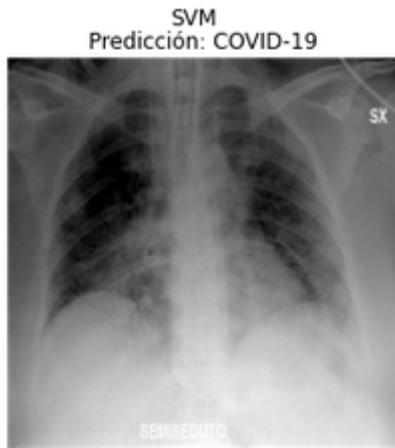
Predicción: Naive Bayes Gaussiano

Imagen con filtro gaussiano y superpíxeles

Gaussiano Naive Bayes
Predicción: COVID-19



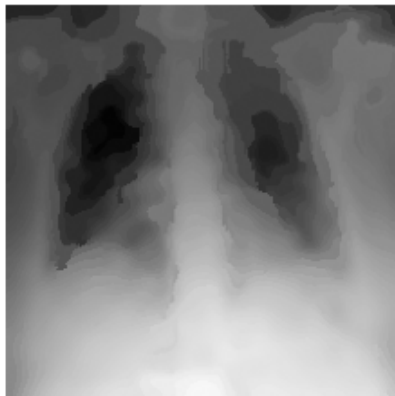
Predicción: SVM



Predicción: SVM

Imagen con filtro gaussiano y superpíxeles

SVM
Predicción: COVID-19



Predicciones

Posteriormente, continuamos con la imagen de rayos X del tórax de una persona normal, de igual manera, se obtuvo la imagen real y la imagen aplicando filtro gaussiano y superpíxeles:

Predicción: KNN

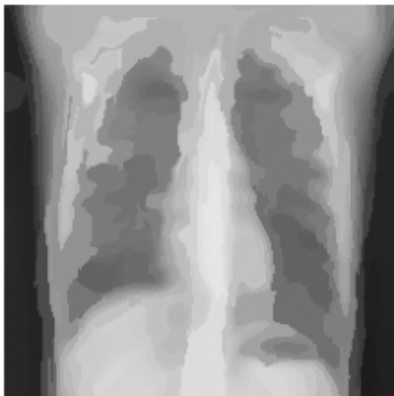
KNN
Predicción: Normal



Predicción: KNN

Imagen con filtro gaussiano y superpíxeles

KNN
Predicción: Normal



Predicción: Naive Bayes Multinomial

Multinomial Naive Bayes
Predicción: Normal

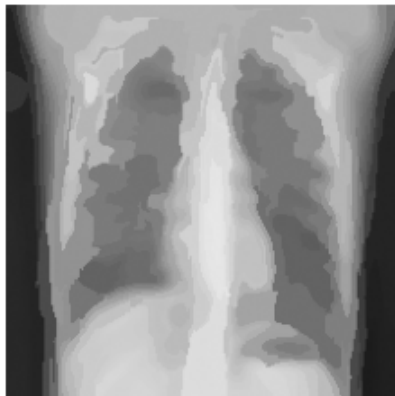


Predicción: Naive Bayes Multinomial

Imagen con filtro gaussiano y superpíxeles

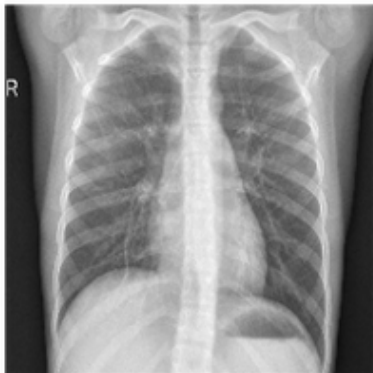
Multinomial Naive Bayes

Predicción: Normal



Predicción: Naive Bayes Bernoulli

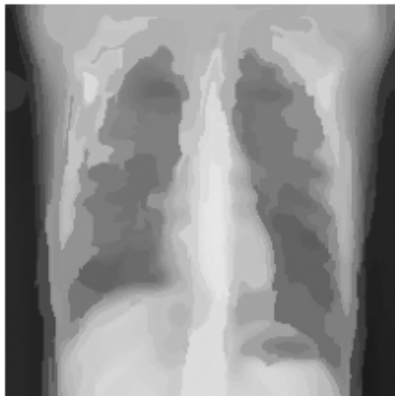
Bernoulli Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Bernoulli

Imagen con filtro gaussiano y superpíxeles

Bernoulli Naive Bayes
Predicción: COVID-19



Predicción: Naive Bayes Gaussiano

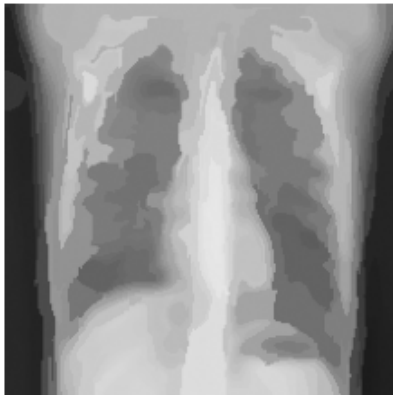
SVM
Predicción: Normal



Predicción: Naive Bayes Gaussiano

Imagen con filtro gaussiano

Gaussiano Naive Bayes
Predicción: Normal



Predicción: SVM

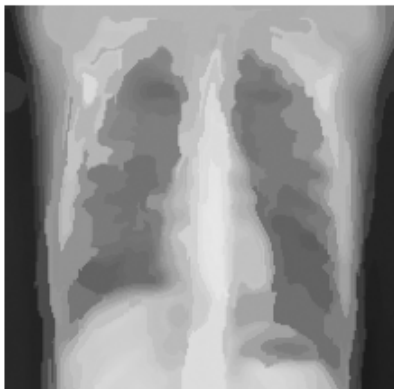
Gaussiano Naive Bayes
Predicción: Normal



Predicción: SVM

Imagen con filtro gaussiano y superpíxeles

SVM
Predicción: Normal



SEGUNDA PARTE: CLASIFICACIÓN POR TOMOGRAFÍAS AXIALES PULMONARES

INDIVIDUO SIN COVID-19 VS INDIVIDUO CON COVID-19

Sobre los datos

- Los datos de pacientes con COVID-19 fueron extraídos de:
<http://medicalsegmentation.com/covid19/>
- Para las imágenes de pacientes que no presentaban COVID-19 se realizaron búsquedas independientes en google.

Sobre los datos

TC axial pulmonar con COVID



TC axial pulmonar sin COVID

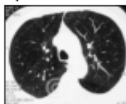


Ajustando imágenes sin COVID

Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)



Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)

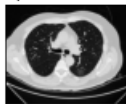
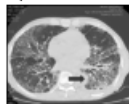
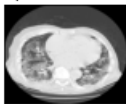


Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)

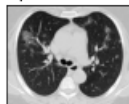
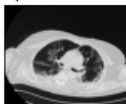


Ajustando imágenes con COVID

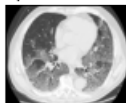
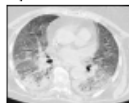
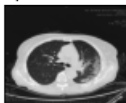
Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)



Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)



Shape: (188, 233) Shape: (188, 233) Shape: (188, 233)



Creando Conjuntos

- El conjunto de valores X.
Cada renglón corresponde a una imagen aplanada.
- El conjunto de valores Y.
Cada renglón corresponde a la etiqueta del conjunto al que pertenece
- Se trabajó con 64 imágenes por conjunto.
- Todos los clasificadores trabajaron con los mismos conjuntos.

KNN; 6 vecinos

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	1.00	1.00	13
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Resultados KNN

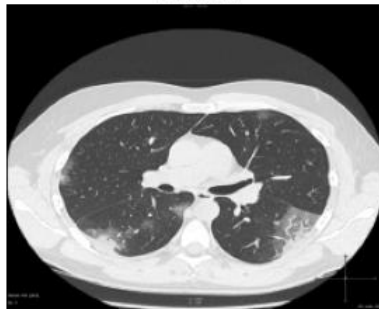
Se eligió Sin Covid

Sin Covid



Se eligió Con Covid

Con Covid



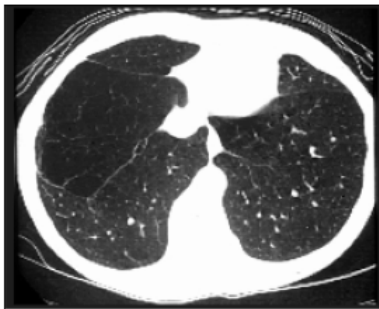
K means; K=2

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	1.00	1.00	13
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Resultados K-means

Se eligió Sin Covid

Sin Covid



Se eligió Con Covid

Con Covid



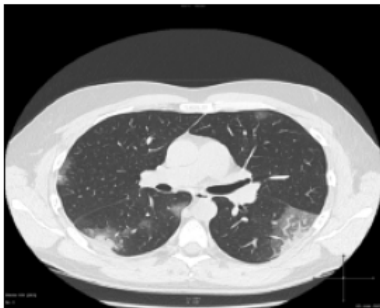
Bayes

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	1.00	1.00	13
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Resultados Bayes

Se eligió Con Covid

Sin Covid



Se eligió Sin Covid

Sin Covid



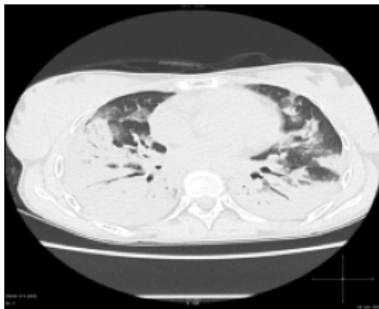
SVM

	precision	recall	f1-score	support
0	1.00	1.00	1.00	19
1	1.00	1.00	1.00	13
accuracy			1.00	32
macro avg	1.00	1.00	1.00	32
weighted avg	1.00	1.00	1.00	32

Resultados SVM

Se eligió Con Covid

Con Covid



Se eligió Con Covid

Sin Covid



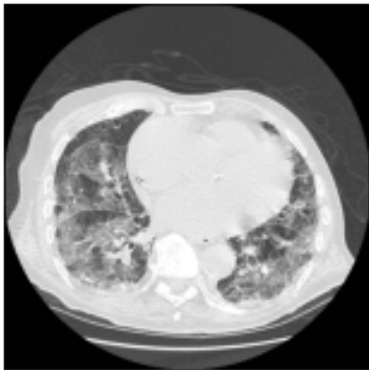
TERCERA PARTE: CLASIFICACIÓN INTERNA INDIVIDUO CON COVID-19

Sobre los datos

- Los datos de pacientes con COVID-19 fueron extraídos de:
<http://medicalsegmentation.com/covid19/>
- Para este caso nos apoyaremos de las máscaras.

Sobre los datos

Imagen 1



Máscara 1



Regiones y distribuciones.

4 regiones.

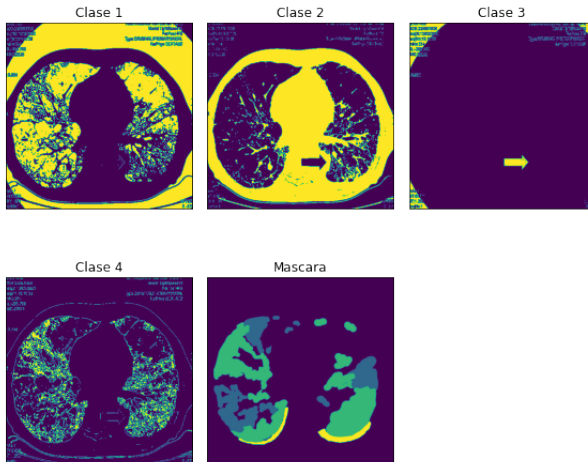
Mayor distancia -> 24,360,199.

- La region 1 cuenta con 24394464 pixeles
- La region 2 cuenta con 1196461 pixeles
- La region 3 cuenta con 589210 pixeles
- La region 4 cuenta con 34265 pixeles

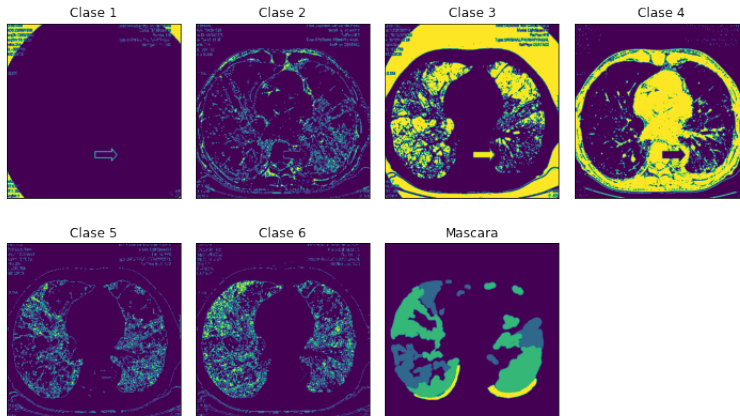
Creando conjuntos

- Se equilibraron los datos a 34,265 pixeles por región.
- 137,060 Pixeles totales.
- El conjunto de valores X.
cada renglón corresponde a un pixel.
- El conjunto de valores Y.
Cada renglón corresponde a la clasificación del pixel en X.
- 70% entrenamiento.
- 30% Prueba.

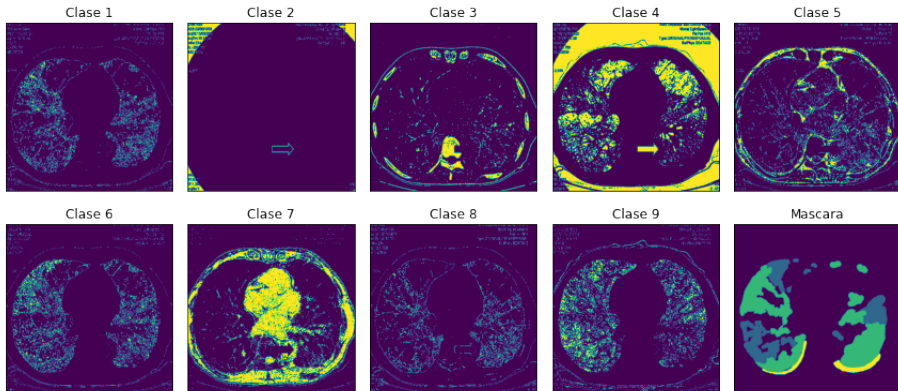
K-means; $K = 4$



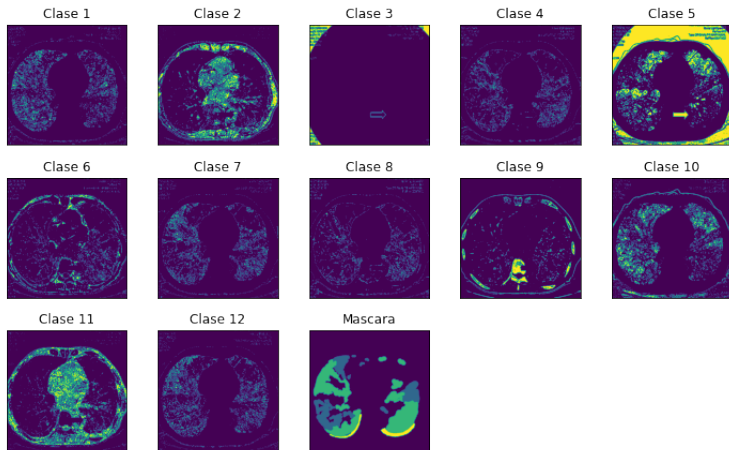
K-means; K = 6



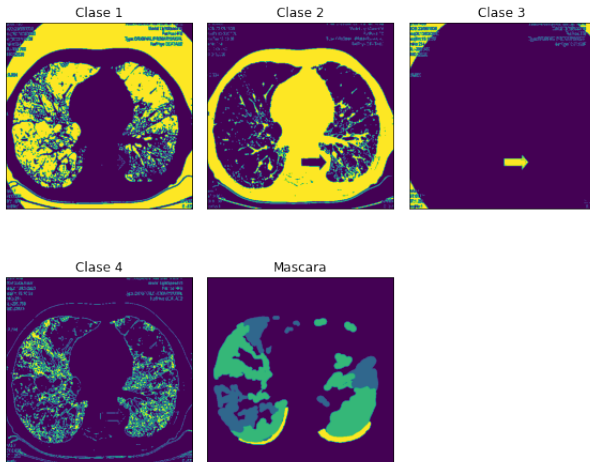
K-means; $K = 8$



K-means; K = 12

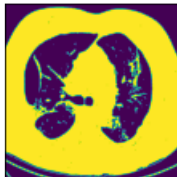


Decisión K=4



Clasificación K-means

Consolidación



Vidrio Esmeralizado

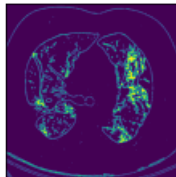
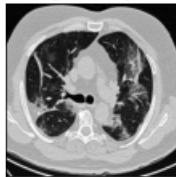


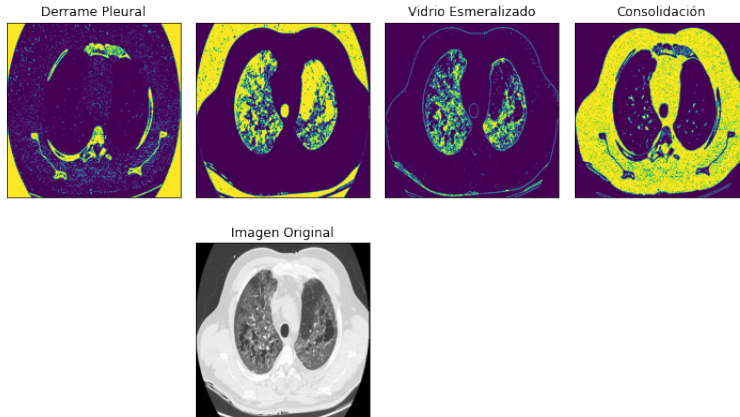
Imagen Original



KNN; 700 vecinos

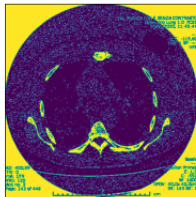
	precision	recall	f1-score	support
0	0.90	0.61	0.73	10258
1	0.57	0.66	0.61	10266
2	0.57	0.57	0.57	10297
3	0.71	0.85	0.77	10297
accuracy			0.67	41118
macro avg	0.69	0.67	0.67	41118
weighted avg	0.69	0.67	0.67	41118

Zonas KNN

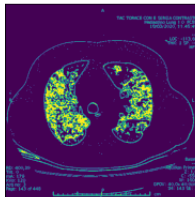


Clasificación KNN

Derrame Pleural



Vidrio Esmeralizado



Consolidación

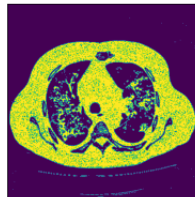


Imagen Original

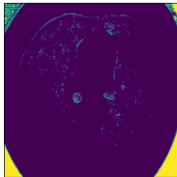


Bayes

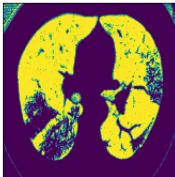
	precision	recall	f1-score	support
0	0.98	0.55	0.71	10258
1	0.55	0.75	0.64	10266
2	0.63	0.38	0.47	10297
3	0.62	0.92	0.74	10297
accuracy			0.65	41118
macro avg	0.70	0.65	0.64	41118
weighted avg	0.70	0.65	0.64	41118

Zonas Bayes

Derrame Pleural



Vidrio Esmeralizado



Consolidación

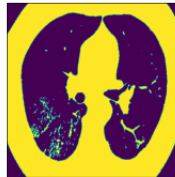
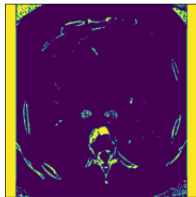


Imagen Original

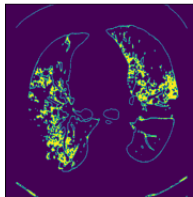


Clasificación Bayes

Derrame Pleural



Vidrio Esmeralizado



Consolidación

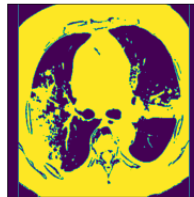


Imagen Original

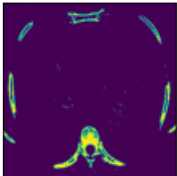


SVM

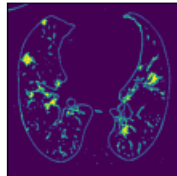
	precision	recall	f1-score	support
0	0.97	0.57	0.72	10258
1	0.58	0.67	0.62	10266
2	0.59	0.58	0.59	10297
3	0.69	0.88	0.77	10297
accuracy			0.68	41118
macro avg	0.71	0.68	0.67	41118
weighted avg	0.71	0.68	0.67	41118

Zonas SVM

Derrame Pleural



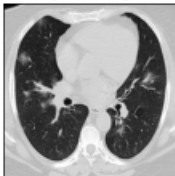
Vidrio Esmeralizado



Consolidación

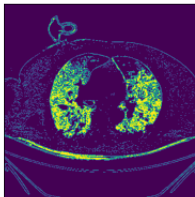


Imagen Original



Clasificación SVM

Vidrio Esmeralizado



Consolidación

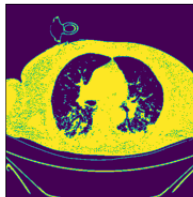
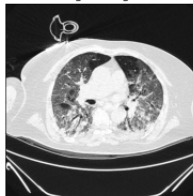


Imagen Original



Conclusiones

- La pandemia del coronavirus ha tomado por sorpresa a todo el mundo, afectando a muchos países, principalmente en lo económico y en cuanto a la salud de las personas. Por ello, a pesar de tomar las correspondiente medidas para evitar la mayor cantidad de contagios posibles, se deben implementar herramientas confiables que faciliten la detección de esta enfermedad, y por eso se realizó este proyecto, para comparar el rendimiento de los clasificadores y ver cuál nos podría ayudar de mejor manera a detectar casos potenciales de coronavirus.
- No se pudieron implementar algunos de los métodos vistos en clases debido a la complejidad de las imágenes. Así mismo, lo que en apariencia podría ser manejado como "ruido" y limpiado con filtros, no se recomienda llevar a cabo, pues resultaba en características importantes dentro del análisis del esmeralizado o la consolidación.

Conclusiones

- Tal como podemos observar al momento de suavizar mediante la aplicación del filtro gaussiano, en algunos algoritmos, llegamos a perder precisión.
- Para el caso de los rayos-X del tórax, pudimos observar que tenemos un buen rendimiento en los algoritmos, aunque la mejora es mayor cuando no realizamos filtro o usamos superpíxeles. Además, los algoritmos que mejor puntaje en la evaluación obtuvieron fueron los de Gauss y SVM, por lo que recomendaríamos esos para clasificar futuras imágenes del tórax de las personas, sobretodo si dependiera de cuidar la salud de las personas que padecen esta enfermedad, pues como mencionamos anteriormente, el clasificador Bernoulli no lo hace correctamente para personas que no padecen la enfermedad.

Conclusiones

- En cuanto a los radiografías de los pulmones, podemos decir que tenemos buenos clasificadores. Todos coinciden en el mismo problema: la consolidación.
- Para identificación de pacientes con COVID a partir de tomografías: KNN y Kmeans tienen un buen rendimiento para identificar de qué tipo de paciente se trata. Sin embargo, se debería trabajar con más imágenes, 64 por cada conjunto es realmente poco.
- Para la identificación de secciones fue un tanto cuanto complicado incluso para nosotros entender cuál era el área consolidada. Como pudimos apreciar, no todos los clasificadores poseen la "habilidad" para identificar correctamente las regiones. SVM es muy bueno, desgraciadamente el tiempo que tarda en clasificar una sola imagen es realmente tardado. En este caso, Bayes aparenta no tener problemas.

!!!Gracias por su atención!!!