UNIVERSIDAD DE ANTIOQUIA

Despliegue de un modelo de clasificación de tumores de cáncer de mama

Reto Kaggle

Juan Camilo Peña Vahos

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de

Especialista en Analítica y Ciencia de Datos

Asesor:

Sebastián Rodríguez Colina

Universidad de Antioquia

Facultad de Ingeniería – Departamento de Ingeniería de Sistemas

Medellín, Colombia

2022

Cita	(Vahos Camilo, 2022)
Referencia **Estilo APA 7 (2020)**	Vahos Camilo (2022). *Despliegue de un modelo de clasificación de tumores de cáncer de mama, Reto Kaggle* [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte III.

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería CENDOI

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: Jhon Jairo Arboleda Céspedes
Decano/Director: Jesús Francisco Vargas Bonilla
Jefe departamento: Diego José Luis Botia Valderrama

Tabla de contenido

Resumen

El cáncer de mama es uno de los tipos de cáncer mas comunes en el mundo y con una tasa de mortalidad bastante alta dado que suele ser detectado en etapas donde los tratamientos no son tan efectivos para combatir la enfermedad. Durante el proceso de diagnóstico es posible hacer uso de modelo de aprendizaje de máquinas que permitan determinar la naturaleza del tumor (benigno o maligno) sin necesidad de los métodos más invasivos como la biopsia. En el mundo ya existen empresas dedicadas a desarrollar este tipo de modelos, que a partir de las métricas del tumor o de imágenes tratan de clasificar el tumor en alguna de estas categorías. Este proyecto parte de un dataset que contiene las métricas de tumores de mama en mas de 500 pacientes. Con esta información se pretende desarrollar un modelo de clasificación que sea competitivo en el mercado actual y que sirva para cortar el proceso de diagnóstico de modo que se reduzca el tiempo del mismo.

1. Análisis Exploratorio de los Datos

1.1 Problema de Negocio

El cáncer de mama es el tipo de cáncer más común en el mundo con cerca de 2,2 millones de casos en el año 2020 (Cancer.net, 2022). Sin embargo, sigue siendo un tipo de cáncer cuya detección y diagnóstico tardío es el factor principal de la mortalidad por la misma enfermedad. Uno de los problemas principales de los procedimientos de detección actual del cáncer son los tiempos para la clasificación del tumor, las pruebas más rápidas suelen ser las más costosas y por tanto la enfermedad suele complicarse a las personas que no tienen acceso a los recursos para una prueba de estas. Es necesario entonces desarrollar un método de diagnóstico rápido y de bajo costo que permita a los profesionales de la salud establecer rápidamente el camino a seguir a un paciente clasificando el tumor como benigno o maligno. Para el diagnóstico completo de un tumor de cáncer de mamá deben ocurrir tres cosas, primero el paciente debe descubrir el tumor o cualquier indicio de una malformación en el área (Para esto se instruyen a las personas para detectar signos de la enfermedad) (Ministerio de Salud, 2020). Segundo se debe hacer una mamografía, ecografía o resonancia magnética, estas tres técnicas permiten tomar ciertas medidas del tumor a través de imágenes. Finalmente, y a consideración del médico, se hace una biopsia que es la que determina si el tumor es benigno o maligno. El objetivo principal es reducir la necesidad de aplicar el paso 3 que es donde radica el mayor costo del proceso (Organización Mundial de la Salud, 2021).

El modelo de clasificación desarrollado recibe como entrada la descripción del tumor que incluye todas las estadísticas que se pueden extraer del mismo. Con esta descripción el modelo se encarga de clasificar el tumor en dos categorías, Maligno o Benigno.

Los datos se obtuvieron a través de la plataforma Kaggle, donde se referencia que provienen de alguna toma de muestras en Wisconsin. El dataset contiene los diagnósticos de más de 500 pacientes, todos ellos clasificados por la naturaleza del tumor (Benigno o Maligno).

1.2 Métricas de Desempeño

La métrica que se va a utilizar para medir el desempeño del modelo aplicado es la exactitud (Accuracy) que se define como la proporción entre el número de predicciones correctas sobre el número total de predicciones. Para este problema se define lo siguiente (Mishra, 2018):

- Verdadero Positivo (TP): Aquellos tumores que se clasifican correctamente como benignos.
- Verdadero Negativo (TN): Aquellos tumores que se clasifican correctamente como malignos.
- Falso Positivo (FP): Aquellos tumores que se clasifican incorrectamente como benignos.
- Falso Negativo (FN): Aquellos tumores que se clasifican incorrectamente como malignos.

Nótese que la exactitud no distingue entre los tipos de aciertos del modelo (verdaderos positivos y verdaderos negativos). Se define la exactitud de la siguiente manera:

$$Exactitud = \frac{Número\ de\ predicciones\ correctas}{Número\ total\ de\ predicciones}$$

Por lo tanto, se usarán la sensibilidad (sensitivity) que también se conoce como la tasa de verdaderos positivos y la especificidad (specificity) que también se conoce como la tasa de verdaderos negativos y que se definen con las siguientes ecuaciones:

$$Sensibilidad = \frac{TP}{FN + TP}$$

$$Especificidad = \frac{TN}{FP + TN}$$

Para seleccionar estas métricas se investigó sobre clasificadores de tumores de cáncer de mama en estado productivo, entre los encontrados está Syantra (Syantra, n.d.), el cual sería competidor directo y además el único que presenta sus métricas. Las métricas de Syantra para mujeres menores de 50 años son 98.5% de precisión, 99% de Especificidad, 91.7% de Sensibilidad. Esta referencia no solamente nos sirve para identificar las métricas con las que vamos a medir el modelo y los resultados, también nos da valores objetivo a los que el proyecto debe tratar de igualar o superar. Se busca mejorar la especificidad para aumentar los casos en que el tumor se clasifica correctamente como benigno mientras se reduce el número de tumores clasificados como malignos cuando en realidad eran benignos. Por otro lado, se busca mejorar la sensibilidad para detectar correctamente los tumores malignos y reducir el número de diagnósticos en el que el tumor se clasifica como benigno cuando en realidad es maligno. La Exactitud es de un carácter más general y orientado a medir el desempeño general del modelo al saber cuantos aciertos de clasificación se obtuvieron en total. El modelo seleccionado será aquel cuyas métricas igualen o superen a las de Syantra, debido a que estas garantizan que se reduce significativamente la necesidad de que los usuarios tengan que recurrir a los estudios de biopsia para clasificar el tumor.

2. Datos

El dataset original (aquí) se encuentra en formato CSV, tiene un total de 569 filas y cada una corresponde a las métricas del tumor de un paciente. Cada fila tiene 31 columnas y no existen datos nulos ni de ningún tipo inválido. La siguiente tabla presenta las variables del dataset con su descripción y rangos.

Figura 1

Descripción y rango de las características del dataset

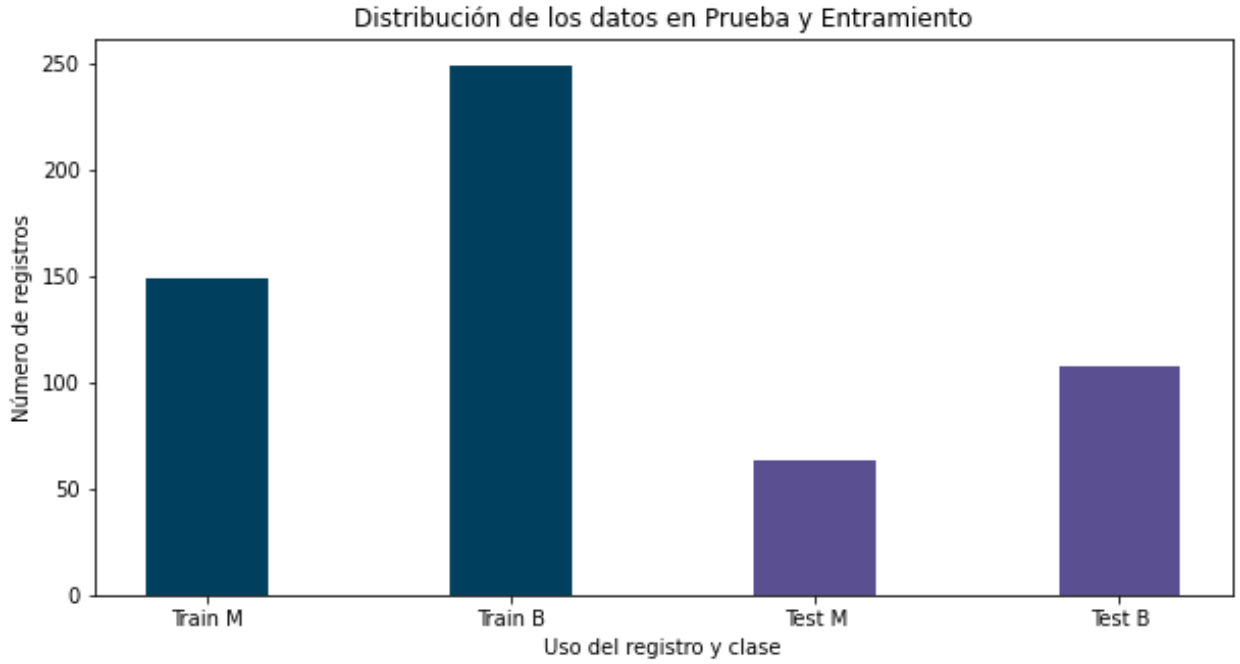
Label	Feature name	Description	Min-max	Mean, ± slandered division
F1	ID number	Integer	—	—
F2	Radius mean	Mean of distances from the center to points on the perimeter cell	6.981000–28.110000	14.127292, ±3.524049
F3	Texture mean	The standard deviation of grayscale values	9.710000–39.280000	19.289649, ±4.301036
F4	Perimeter mean	Perimeter of cell	43.790000–188.500000	91.969033, ±24.298981
F5	Area mean	Area of cell	143.500000–2501.000000	654.889104, ±351.914129
F6	Smoothness mean	Local variation in radius lengths	0.052630–0.163400	0.096360, ±0.014064
F7	Compactness mean	Perimeter ^ 2/area—1.0	0.019380–0.345400	0.104341, ±0.052813
F8	Concavity mean	The severity of concave portions of the contour	0.000000–0.426800	0.088799, ±0.079720
F9	Concave points mean	Number of concave portions of the contour	0.000000–0.201200	0.048919, ±0.038803
F10	Symmetry mean	Symmetry	0.106000–0.304000	0.181162, ±0.027414
F11	Fractal dimension mean	Coastline approximation"—1	0.049960–0.097440	0.062798, ±0.007060
F12	Radius severity	—	0.111500–2.873000	0.405172, ±0.277313
F13	Texture severity	—	0.360200–4.885000	1.216853, ±0.551648
F14	Perimeter severity	—	0.757000–21.980000	2.866059, ±2.021855
F15	Area severity	—	6.802000–542.200000	40.337079, ±45.491006
F16	Smoothness severity	—	0.001713–0.031130	0.007041, ±0.003003
F17	Compactness severity	—	0.002252–0.135400	0.025478, ±0.017908
F18	Concavity severity	—	0.000000–0.396000	0.031894, ±0.030186
F19	Concave points severity	—	0.000000–0.052790	0.011796, ±0.006170
F20	Symmetry severity	—	0.007882–0.078950	0.020542, ±0.008266
F21	Fractal dimension severity	—	0.000895–0.029840	0.003795, ±0.002646
F 22	Radius worst	—	7.930000–36.040000	16.269190, ±4.833242
F23	Texture worst	—	12.020000–49.540000	25.677223, ±6.146258
F 24	Perimeter worst	—	50.410000–251.200000	107.261213, ±33.602542
F 25	Area worst	—	185.200000–4254.000000	880.583128, ±569.356993
F26	Smoothness worst	—	0.071170–0.222600	0.132369, ±0.022832
F27	Compactness worst	—	0.027290–1.058000	0.254265, ±0.157336
F28	Concavity worst	—	0.000000–1.252000	0.272188, ±0.208624
F29	Concave points worst	—	0.000000–0.291000	0.114606, ±0.065732
F30	Symmetry worst	—	0.156500–0.663800	0.290076, ±0.061867
F31	Fractal dimension worst	—	0.055040–0.207500	0.083946, ±0.018061
y	Diagnosis	M = malignant = 1B = benign = 0	—	—

Los datos tienen un peso de 125 KB , lo que nos permite manipularlos fácilmente en cualquier ambiente, sea local o en la nube. No se tiene ninguna restricción de acceso a los datos puesto que se obtuvieron de la plataforma pública Kaggle.

Para la generación de los distintos datasets se utilizó la función *train_test_split* de la librería *sci-kit learn*. Esta utilidad permite separar los datos disponibles en el subconjunto de pruebas y el subconjunto de entrenamiento de manera aleatoria. Los datasets de prueba y entrenamiento se definieron en el primer paso de todas las iteraciones para garantizar que los modelos no estuvieran sesgados. La distribución de los dataset se presenta en la siguiente tabla.

Figura 2

Distribución del dataset original en prueba y entrenamiento después del *train_test_split*



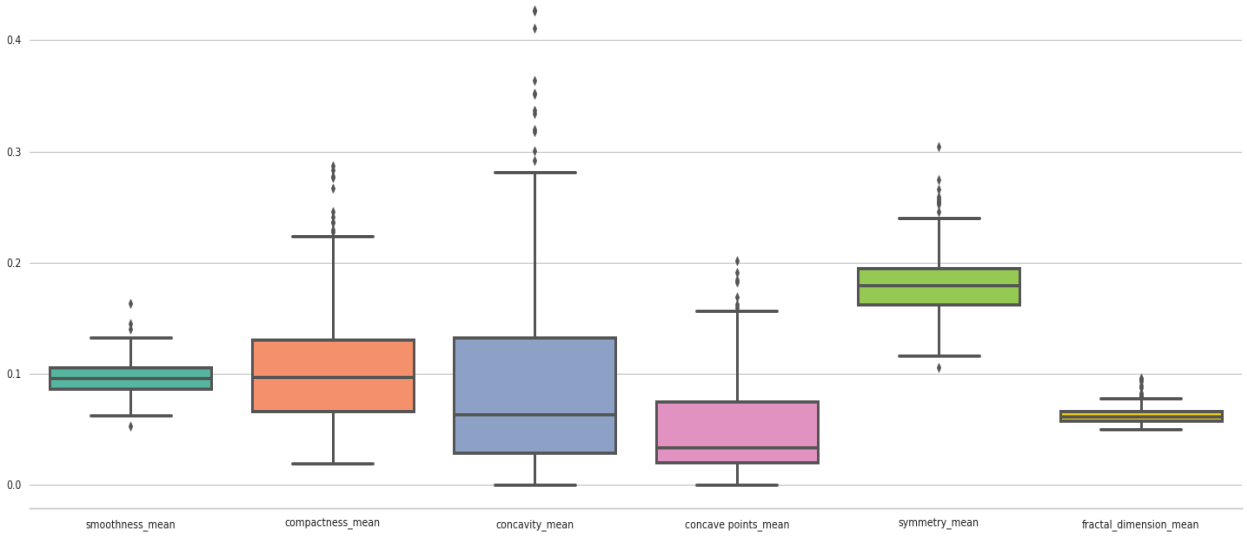
2.1 Analítica Descriptiva

El dataset de entrenamiento contiene 198 registros, cada uno con 31 características. Entre las características categóricas se tiene la identificación del paciente (id) la cual se ha eliminado por no aportar ni en el entrenamiento ni en la clasificación. Finalmente, la variable de salida es el

diagnóstico (diagnosis) el cual toma dos valores B (benigno) y M de (maligno) indicando la naturaleza del tumor. El dataset no contiene datos nulos ni registros incompletos por lo que solo hace falta identificar la presencia de datos atípicos. En primer lugar se hizo uso de diagramas de caja y bigotes para identificar si las diferentes características del dataset contenían datos atípicos, el resultado se puede evidenciar en las siguientes figura. En general se puede notar que algunas de las características presentan datos atípicos por lo que se decidió implementar el algoritmo *Local Outlier Factor* que también se encuentra en la librería *scikit learn.*

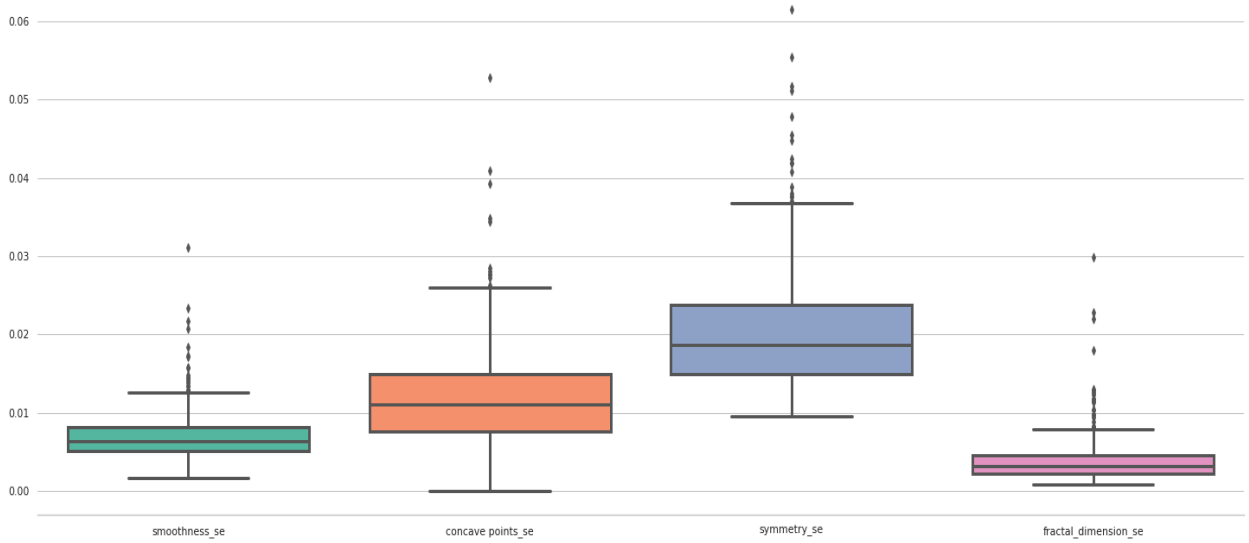
Figuras 3

Diagramas de caja y bigotes para la detección de datos atípicos



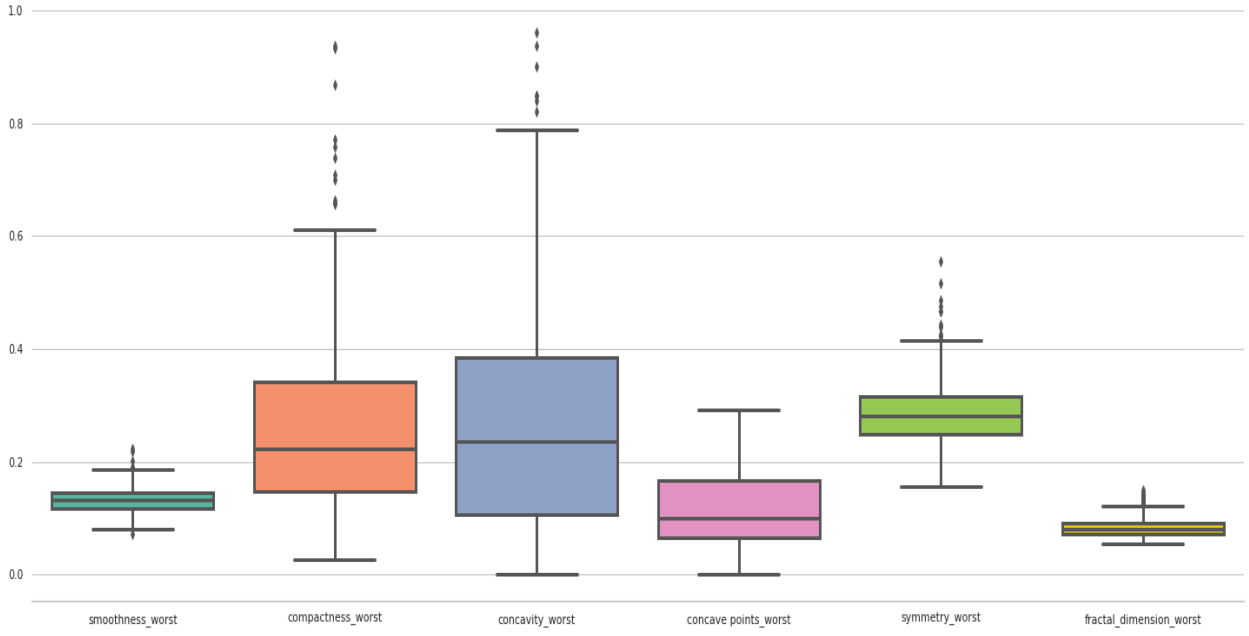
Figuras 4

Diagramas de caja y bigotes para la detección de datos atípicos



Figuras 5

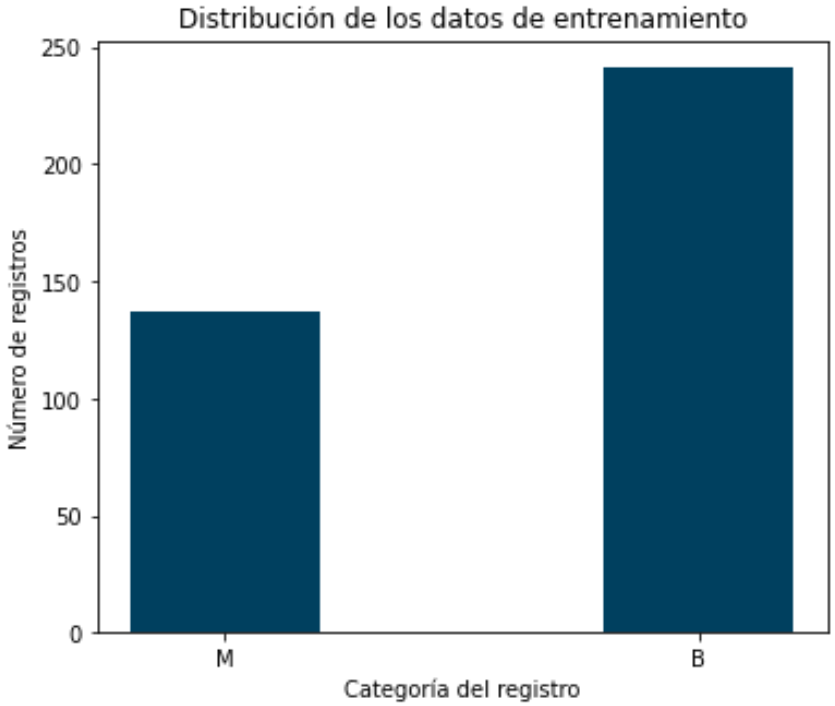
Diagramas de caja y bigotes para la detección de datos atípicos



El algoritmo detectó un total de 20 filas con datos atípicos, los cuales se eliminaron con el fin de reducir la variabilidad en los datos y dejar solo aquellos que son estadísticamente significativos. La nueva distribución del dataset de entrenamiento es la siguiente.

Figura 6

Distribución de los datos de entrenamiento después de eliminar los datos atípicos



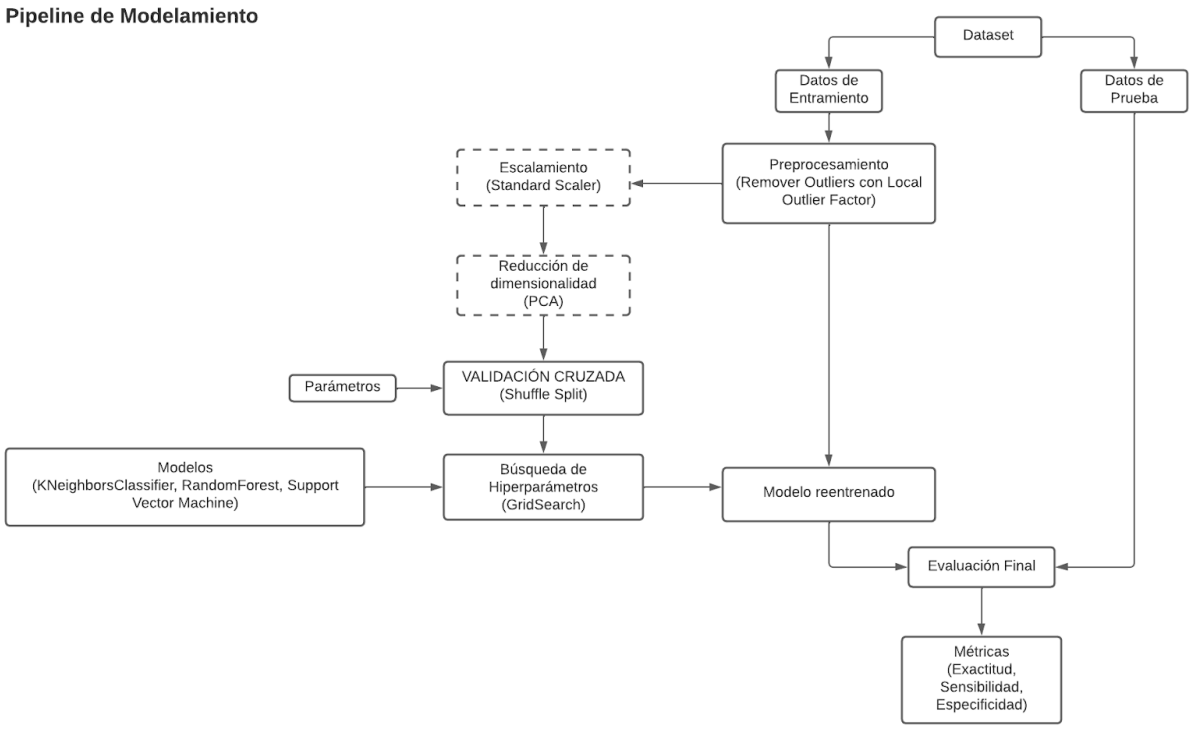
3. Proceso de Analítica

3.1 Pipeline Principal

El siguiente diagrama presenta los pasos a seguir para cada una de las iteraciones con los modelos propuestos para solucionar el problema.

Figura 7

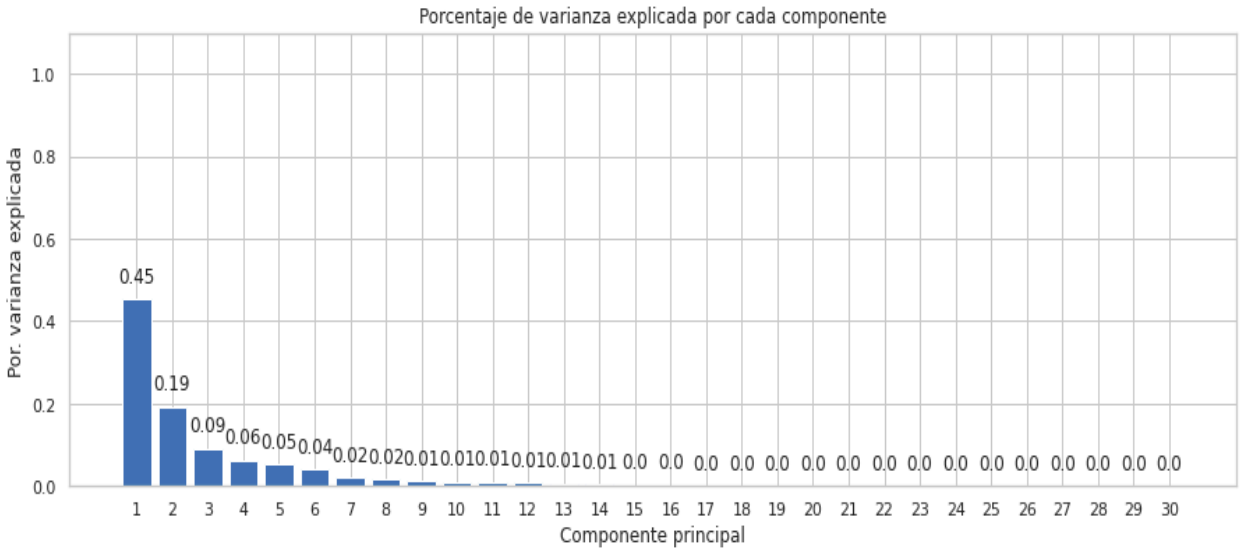
Pipeline principal



En el pre-procesamiento se eliminan los datos atípicos como se evidenció la sección anterior. Después de esto se aplica una reducción de dimensionalidad al dataset de entrenamiento y un escalamiento dependiendo del modelo que se vaya a aplicar, es decir, algunos modelos omiten el escalamiento ya que no es necesario. La reducción de dimensionalidad se hace a través del algoritmo PCA. La siguiente figura presenta el porcentaje de varianza que explica cada una de las características del dataset una vez aplicado PCA.

Figura 8

Porcentaje de la varianza explicada por cada carácterística



De acuerdo con la figura anterior, la reducción de dimensionalidad con la técnica PCA, se aplica para reducir a 7 características, esto es porque el 90% de la varianza se explica con 7 variables. La reducción de dimensionalidad no es una etapa obligatoria del pipeline, solo se aplicó para evaluar el desempeño de los modelos si se reduce el número de características, aunque el tamaño del dataset no sea un aspecto crítico en nuestro caso. Sin embargo, como se verá mas adelante, el modelo con PCA tiene pérdidas en las métricas y por tanto no se tendrá en cuenta en la fase de producción.

3.2 Modelos

El problema que se enfrenta aquí es binario, es decir que solo tenemos que clasificar una entrada en 2 posibles clases, en este caso, si el tumor es Benigno o Maligno; además, las clases se pueden considerar como balanceadas. La siguiente tabla presenta los modelos aplicados y los parámetros evaluados para cada modelo, cabe resaltar que la evaluación de los modelos se hizo para los datos con y sin PCA aplicado.

Tabla 1. Modelos aplicado y parámetros a evaluar por modelo

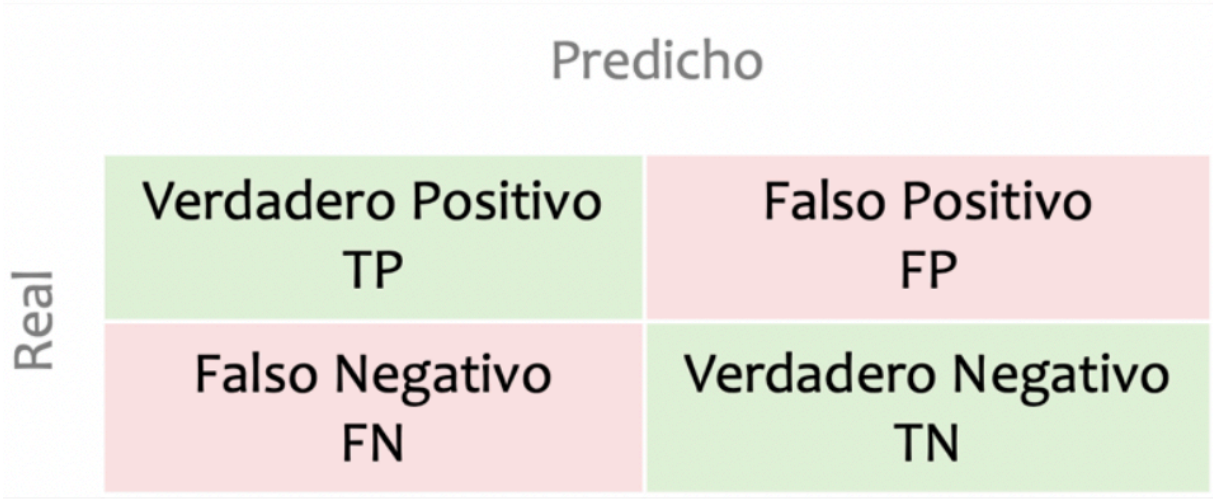
Algoritmos de Clasificación	Parametrización
KNeighborsClassifier	• K: 1, 3, 5, 7, 9, 11 • Metrics: euclidian, manhattan, chebyshev
Random Forest Classifier	• Bootstrap: True • Max_depth: 80, 90, 100, 110 • Max_features: 2, 3 • Min_samples_leaf: 3, 4, 5 • Min_samples_split: 8, 10, 12 • N_estimators: 100, 200, 300, 1000
Support Vector Machine	• Kernel: linear

3.3 Métricas

Para cada uno de los modelos se calcularon las métricas exactitud (accuracy), sensibilidad (sensitivity) y especificidad (specificity). Partiendo de la matriz de confusión la cual nos muestra los resultados de la evaluación del modelo tal y como se muestra en la siguiente figura.

Figura 9

Distribución de la matriz de confusión



Para el cálculo de esto se usaron las funciones *accuracy_score* y *confusion_matrix,* ambas pertenecientes a la librería *scikit learn*. La función *accuracy_score* recibe como parámetro la lista con los valores correctos de cada uno de los datos de prueba (*y_test*) y la lista de valores predichos

por el modelo (*y_pred*), con estos dos valores cálcula la exactitud del modelo. Con la función confusion_matrix se puede obtener los valores de *TP, FP, FN y TN*. Una vez se tienen esos valores se puede calcular la especificidad y la sensibilidad siguiendo las ecuaciones descritas anteriormente para estas métricas.

4. Metodología

4.1 Baseline

Para el baseline se asumió que los datos presentan una distribución de Bernoulli. Partiendo del dataset de entrenamiento preprocesado se calcula el valor de p para una distribución binomial así:

$$p = \frac{241}{378} = 0.6375$$

$$q = 1 - p = \frac{137}{378} = 0.3625$$

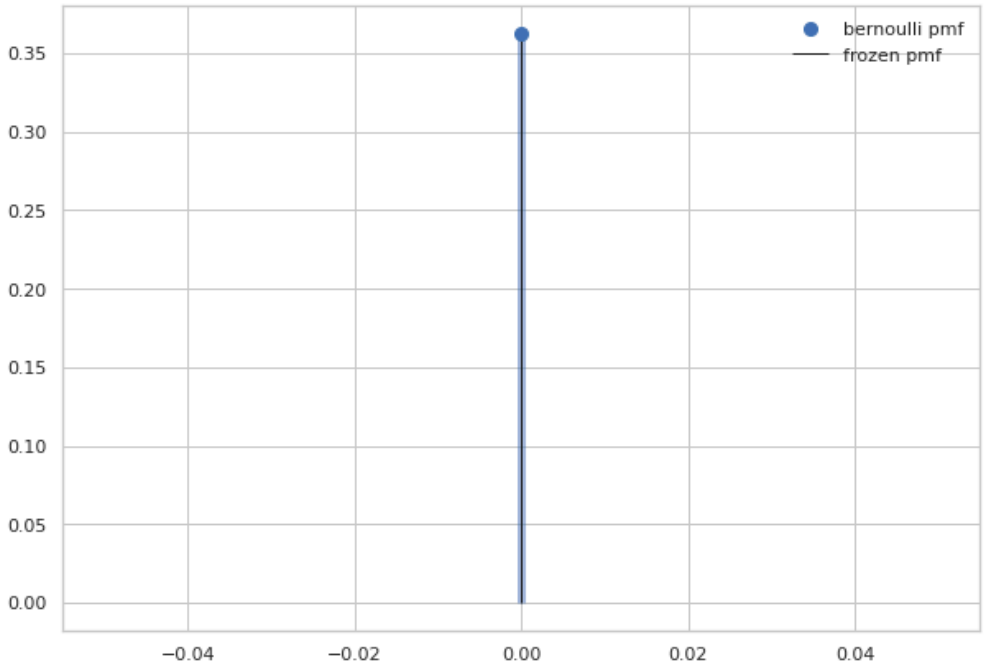
Por tanto

$$\mu = p;\ \ \sigma = pq\ \ = 0.231$$

Con estos valores se modela una distribución de Bernoulli cuya PDF se puede observar en la siguiente gráfica

Figura 10

PDF de la distribución Bernoulli de los datos



Para esta distribución calculamos las métricas y se obtuvo que:

$$Specificity = 0.537$$

$$Sensibility = 0.444$$

$$Accuracy = 0.503$$

4.2 Validación

Los experimentos se validan usando dos herramientas. La primera de ellas es la validación cruzada usando *ShuffleSplit*. Esta técnica permite realizar varios entrenamientos y pruebas al dividir los datos de entrenamiento nuevamente en dos subconjuntos, uno de entrenamiento y otro de prueba, pero esto lo hace en varias iteraciones y dividiendo los datos siempre con valores diferentes en cada subconjunto; de esta forma se puede evitar que el modelo sea entrenado con sobreajuste y que tampoco tenga sesgos de selección. La segunda validación es para los hiperparámetros de cada modelo, en este caso se usa la técnica *GridSearch* que permite explorar todas las combinaciones de hiperparámetros de modo que se seleccioné al final aquellos que registraron el mejor resultado en una métrica en particular que se denomina *scoring*; en este caso, se usó la exactitud como métrica de evaluación. Una vez se hacen estas dos validaciones, se entrena el modelo con los mejores parámetros y se usa el dataset de pruebas para hacer la evaluación final y obtener las métricas de desempeño del modelo seleccionado anteriormente.

5. Resultados

5.1 Métricas

Los resultados se presentan en las siguientes tablas para los 3 modelos evaluados con y sin PCA aplicado en el pipeline.

Tabla 2. Resultados del Baseline

MODELO	EXACTITUD	SENSIBILIDAD	ESPECIFICIDAD
Baseline	0.503	0.444	0.537

Tabla 3. Resultados para la métrica Exactitud

MODELO	Exactitud	
	Con PCA	Sin PCA
KNeighbors	0.947	0.959
Random Forest	0.923	0.935
Support Vector Machine	0.964	0.953

Tabla 4. Resultados para la métrica Sensibilidad

MODELO	Sensibilidad	
	Con PCA	Sin PCA
KNeighbors	0.905	0.905
Random Forest	0.888	0.889
Support Vector Machine	0.952	0.937

Tabla 5. Resultados para la métricas Especificidad

MODELO	Especificidad	
	Con PCA	Sin PCA
KNeighbors	0.972	0.991
Random Forest	0.944	0.963
Support Vector Machine	0.972	0.963

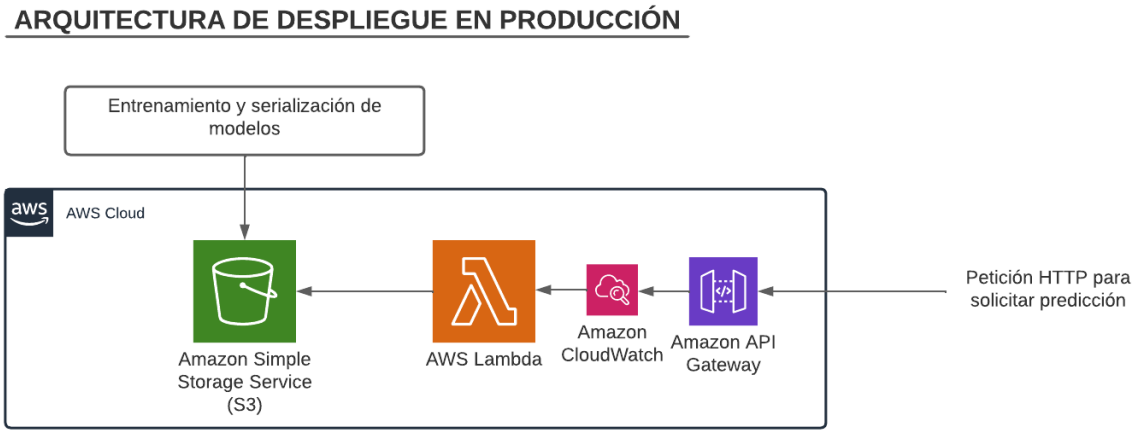
En general, los resultados de todos los modelos con y sin PCA fueron similares; sin embargo, se puede evidenciar que al implementar PCA los modelos bajaban sus métricas en todos

los casos. Esto puede deberse a que el PCA se implementó para reducir el conjunto a 7 características que explicaban el 90% de la variabilidad de los datos, es posible que el 10% faltante sea el que mejore la clasificación cuando se aplica el modelo sin PCA. Es claro que Random Forest no es un candidato opcionado para ser usado en producción, además de que presentó el mayor costo computacional, también entregó los peores resultados en las métricas. Los modelos SVC y KNeighbors presentan métricas muy similares, uno de ellos presenta mejores resultados en la Especificidad y el otro en la Sensibilidad. Al final, el más opcionado para ser usado en la puesta en producción es el modelo KNeighbors que tiene la mejor exactitud de todos.

5.2 Consideraciones de Producción

Figura 11

Arquitectura de despliegue en producción



La figura anterior presenta la arquitectura que se usará en la etapa productiva del proyecto. A continuación, se describen cada uno de los componentes de dicha arquitectura.

1. Entrenamiento del modelo: Para el entrenamiento del model se usará una interfaz de línea de comandos (CLI). Esta pieza de software permitirá a un usuario entrenar modelos con diferentes parámetros y subirlos a un contenedor S3 en AWS después de serializarlos

usando la librería joblib que permite guardar objetos únicos de python en archivos a través de la función dump.

2. AWS S3: Es una herramienta de AWS que permita el almacenamiento de cualquier tipo de archivos y su posterior consumo a través de la API de AWS.
3. AWS lambda: Esta herramienta permite ejecutar funciones en la nube usando diferentes disparadores (triggers). En este caso se ejecuta una REST API que recibe como entrada las características del tumor, luego extrae el modelo entrenado de S3 y lo usa para poder hacer una predicción. El valor resultante es devuelto a quien hizo la petición.
4. AWS CloudWatch: Esta herramienta nos permite el monitoreo de nuestro servicio en la nube.
5. AWS API Gateway: Esta herramienta nos permite crear un punto de acceso con rutas para que los usuarios puedan consumir el servicio. Nótese que también hace las veces de disparador de la función lambda que se ejcutará al detectar una nueva petición HTTP en alguna de las rutas expuestas por la API Gateway.

Con esta infraestructura un usuario cualquiera podrá desde un cliente (llámese aplicación web o algún servicio embebido) hacer peticiones enviando las características del tumor y obtendrá como respuesta si el tumor es benigno (B) o maligno (M).

5.3 Costos del Servicio en Producción

Para un estimado de un millón de peticiones al mes se desglosan a continuación los costos del servicio en su etapa productiva:

TABLA 6. Proyección de costos del servicio

SERVICIO	COSTO TRANSACCIÓN	COSTO TOTAL
S3	0.09 USD Gb transferido	62 USD (Teniendo en cuenta que cada transferencia son 125Kb)
Lambda	0,0000166667 USD por cada GB/segundo	0.90 USD
CloudWatch	0,30 USD/mes	0.30 UDS/mes
API Gateway	0,00001345 USD	74,76 USD

En el mundo se estiman alrededor de 2.2 millones de pacientes con cáncer de mamá al año, si asumimos que este valor es un tercio de la población que busca diagnosticar un tumor (Valor aproximado usando la probabilidad binomial calculada anteriormente), tenemos una población de cerca de 6.6 millones de potenciales consumidores de la API. Si cada uno de ellos hace 10 llamados a la API, tenemos un total de 66 millones de peticiones al año que al mes corresponden a 5.5 millones de llamados. Por tanto, el costo mensual del proyecto es de alrededor de 137,96 USD

Finalmente se deja el link con el código del proyecto en Github.

6. Conclusiones

- En la etapa experimental se trabajó con 3 modelos, KNeighbors, Support Vector Machine y Random Forest. El modelo KNeighbors obtuvo el mejor resultado de los 3 mientras que Random Forest el peor.
- Las métricas obtenidas con KNeighbors se acercaron considerablemente a las de Syantra. Para la exactitud se logró un 95.9% mientras que Syantra tiene 98.5%. La sensibilidad de este proyecto es de 90.5% y la Syantra es de 91.7%. Finalmente, la especificidad de ambos es del 99%. Esto se considera como un éxito y una motivación para explorar cómo mejorar el modelo actual.
- La reducción de dimensionalidad con PCA de 31 variables a 7 no mejoró el desempeño del modelo, de hecho, tuvo el efecto contrario y aunque los resultados con PCA son cercanos a los resultados sin PCA, para este caso, la importancia de mejorar las métricas es mayor a cualquier otro beneficio que se pueda obtener del PCA. Por otra parte, para futuras pruebas, se podría intentar aumentando el número de características y evaluar nuevamente las métricas.
- En materia productiva se logró desplegar el modelo en AWS y su consumo a través de una REST API *serverless*. Esto sirve para que el usuario final tenga pueda interactuar fácilmente con el modelo a través de alguna aplicación que consuma los servicios.

7. Referencias

Cancer.net. (2022, January). *Cancer.net*. Retrieved from Cancer.net: https://www.cancer.net/es/tipos-de-cancer/cancer-de-mama/estadisticas

Ministerio de Salud. (2020, Octubre 19). *Minsalud*. Retrieved from Minsalud: https://www.minsalud.gov.co/Paginas/Detecte-el-cancer-de-mama-a-tiempo.aspx#:~:text=De%20acuerdo%20con%20estimaciones%20de,1%20afectadas%20por%20100.000%20habitantes.

Organización Mundial de la Salud. (2021, Marzo 26). *Cáncer de mama*. Retrieved from WHO: https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer

Mishra, A. (2018, Febrero 24). *Metrics to Evaluate your Machine Learning Algorithm*. Retrieved from Towards Data Science: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

Syantra. (n.d.). *Syantra*. Retrieved from Syantra: https://www.syantra.com/for-healthcare-providers#:~:text=Syantra%20DX%20%7C%20Breast%20Cancer%20is%20an%20advanced%2C%20innovative%20precision%20medicine,early%20stages%2C%20before%20it%20spreads.

Scikit Learn. (n.d.). *RandomForestClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Scikit Learn. (n.d.). *Support Vector Machines*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/svm.html#support-vector-machines

Scikit Learn. (n.d.). *KNeighborsClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Scikit Learn. (n.d.). *GridSearchCV*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

Scikit Learn. (n.d.). *Metrics and Scoring*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/model_evaluation.html