

Modelos y Simulación II: Detección y Clasificación de niveles de retinopatía diabética

Proyecto del Curso

Andrés Quintero Bedoya - 1216727950, Juan Carlos Calle - 1039452184, Prof. Julian David Arias Londoño
Departamento de Ingeniería de Sistemas, Universidad de Antioquia, Colombia

Resumen—EL presente proyecto se realiza como parte del desarrollo del curso Modelos y Simulación II de la Universidad de Antioquia. La finalidad principal es aplicar los conocimientos obtenidos en la materia por medio de la aplicación y descripción lo mas completa posible de un proyecto seleccionado. El proyecto escogido es una competencia de Kaggle basado en imágenes oculares de alta resolución sobre la clasificación de los niveles en la retinopatía diabética. El proceso de clasificación consiste en reconocer detalles muy finos como los micro aneurismas en algunas características más grandes como los exudados. Las tecnologías contemporáneas nos permiten desarrollar dispositivos capaces de detectar automáticamente la condición de los ojos de una persona en función de sus imágenes retinianas. Sin embargo, en la práctica varios factores dificultan la calidad de las imágenes capturadas e impiden el resultado de la detección. En el presente trabajo se hace un pequeño estado del arte de varios trabajos realizados en la competencia de Kaggle de la base de datos seleccionada y a su vez se hace una extracción de las características de las imágenes muy básica (con fines educativos). El método ha sido probado en un conjunto de imágenes de la retina de personas que viven en las zonas rurales del sur de Asia, lo que resultó en una precisión de clasificación de aproximadamente el 40 %, es muy poco pero cumple con los objetivos de la materia ya que en lo posible tratamos de usar los métodos aprendidos en clase.

Index Terms—Machine Learning, Modelos matemáticos, Retinopatía Diabética, Sistema de Apoyo al Diagnóstico de Decisiones Clínicas.

I. INTRODUCCIÓN

LA RETINOPATÍA DIABÉTICA (DR) es una de las principales causas de ceguera a nivel mundial. La retinopatía diabética es una afección ocular que afecta a las personas con diabetes. Tres de cada cuatro personas que viven con diabetes (352 millones) están en edad activa (es decir, entre 20 y 64 años). Se prevé que este número aumente a 417 millones para 2030 y a 486 millones para 2045[1]. La retinopatía diabética es una complicación muy temida de la diabetes, además del edema macular diabético (EMD), catarata, glaucoma, visión doble y la incapacidad de enfoque. En muchos países, se sabe que la RD es una de las principales causas de la ceguera en la población de edad activa, con consecuencias devastadoras personales y socio económicas, a pesar de que se puede prevenir y tratar[2]. Una de las formas más comunes de detectar la enfermedad diabética del ojo es que un especialista examine imágenes de la parte posterior del ojo y determine si hay signos de la enfermedad y, de ser así, qué tan grave es.

El objetivo de este trabajo es que utilizando los conocimientos, herramientas y modelos vistos en el curso realizar median-

te diferentes técnicas estadísticas y de Machine Learning (ML) modelos que tomen estas imágenes y generen un diagnóstico donde clasifique el nivel respectivo al que corresponde la imagen.

La declaración del problema abordado es un problema de desafío abierto en una competencia de la pagina de Kaggle[3]. Este problema tiene como objetivo desarrollar un sistema que pueda ayudar a los oftalmólogos a detectar la ceguera antes de que ocurra. El problema seleccionado utiliza el conjunto de datos proporcionado por Kaggle, que son imágenes del ojo de alta resolución y se trata de identificar defectos en el ojo usando cada píxel individual de la imagen para clasificar la imagen. Las imágenes se obtuvieron de EyePACS en los Estados Unidos y de 3 hospitales oftalmológicos en la India (Aravind Eye Hospital, Sankara Nethralaya y Narayana Nethralaya) entre los pacientes que se sometieron a pruebas de detección de retinopatía diabética. El conjunto de datos utilizado consistió en una muestra aleatoria de imágenes centradas en la mácula tomadas en los sitios de detección de EyePACS entre mayo de 2015 y octubre de 2015. Se utilizaron una variedad de cámaras, incluidas Centervue DRS, Optovue iCam, Canon CR1 / DGi / CR2 y Topcon NW utilizando 45 en los campos de visión. Las imágenes de EyePACS se obtuvieron como parte de la atención clínica de rutina para el cribado de la retinopatía diabética, y aproximadamente el 40 % de las imágenes se obtuvieron con dilatación de la pupila[4].

La siguiente tabla representa el porcentaje de cada una de las clases presentes en los datos.

Cuadro I
TABLA DE CLASIFICACIÓN DE LAS IMÁGENES

Clase	Nombre	Numero de Imágenes	Porcentaje
0	Normal	25810	73.48 %
1	Mild NPDR	2443	6.96 %
2	Moderate NPDR	5292	15.07 %
3	Severe NPDR	873	2.48 %
4	PDR	708	2.01 %

Table I, Clasificación de las imágenes

II. DESCRIPCIÓN DE VARIABLES

En la tabla 1 el significado de las variables de clasificación es la siguiente : El nivel 0 (cero) significa no patológico quiere decir no padece retinopatía diabética. El nivel 1 (uno) es Retinopatía Diabética No Patológica (NPDR) leve, significa que se encuentran micro aneurismas (manchas rojas) que son la fuente del exudado duro (manchas amarillas de alto contraste)

a veces en patrones circulares, esta es la etapa mas temprana, donde solo pueden ocurrir micro aneurismas. El nivel 2 (dos) es NPDR moderado es una etapa en la cual se puede perder la capacidad del transporte de la sangre por los vasos sanguíneos debido a su distorsión e hinchazón. El nivel 3 (tres) es NPDR grave, en esta etapa se impide el suministro de sangre hacia la retina debido al bloqueo de vasos sanguíneos. El nivel 4 (cuatro) es Retinopatía Diabética Proliferativa (PDR) es la etapa más avanzada, donde las características de crecimiento, secretadas por la retina activan la proliferación de nuevos vasos sanguíneos, creciendo al interior de la retina en un gel vítreo, llenando el ojo. Cada etapa tiene sus características y propiedades particulares, por lo que los médicos posiblemente no podrían tomar algunos detalles en cuenta, y por lo tanto existe la probabilidad de que hagan un diagnóstico incorrecto.

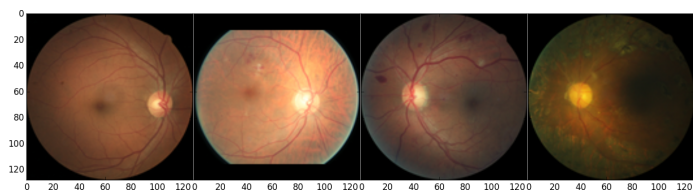


Figura 1. Ejemplos de DB. Ejemplos de clases patológicas 1,2,3,4 de izquierda a derecha.

III. ESTADO DEL ARTE.

Se realizó una búsqueda de 4 artículos que hayan abordado el mismo problema que intentamos solucionar en el presente trabajo. En los criterios de inclusión se trató de que los trabajos definieran bien la metodología usada para resolver el problema o que incluyan los procesos realizados, se escogieron textos publicados en 2015 en adelante, se seleccionan textos que en lo posible sean registros en conferencias, artículos, revistas reconocidas o libros de alta relevancia. También se buscaron artículos que hayan trabajado con la misma base de datos.

El primer artículo seleccionado fue: **“Understanding Deep Neural Network Predictions for Medical Imaging Applications”** [6]. En este artículo, se investiga los resultados proporcionados por las redes neuronales profundas para la detección de malaria, retinopatía diabética, tumor cerebral y tuberculosis en diferentes modalidades de imagen. Inicialmente en la clasificación de las imágenes utilizan CAM de GoogLeNet y ResNet para la detección de DR en imágenes retinianas. En esta investigación, se centraron únicamente en la detección de DR, por lo tanto, fusionaron todos los casos leves, moderados, proliferativos y graves en una sola categoría DR positiva. Dividieron el conjunto de datos en grupos de 72 %, 8 % y 20 % para capacitación, validación y prueba respectivamente. No hay procesamiento previo, excepto convertir estas imágenes al tamaño de entrada de la red. En el documento se estudia los resultados obtenidos utilizando los enfoques basados en el aprendizaje de transferencia con ResNet y GoogLeNet. Las siguientes Figuras presentan las matrices de confusión y las curvas ROC obtenidas utilizando estos enfoques para la detección de DR. La Tabla 2 resume el rendimiento en

términos de precisión general y AUC para la detección de DR utilizando estas redes.

Cuadro II
PRECISIÓN GENERAL Y AUC PARA LA DETECCIÓN DE DR.

Método	Precisión general	AUC
GoogLeNet	97.3	0.9943
ResNet	96.2	0.9939

Table II, Precisión general y AUC para la detección de DR.

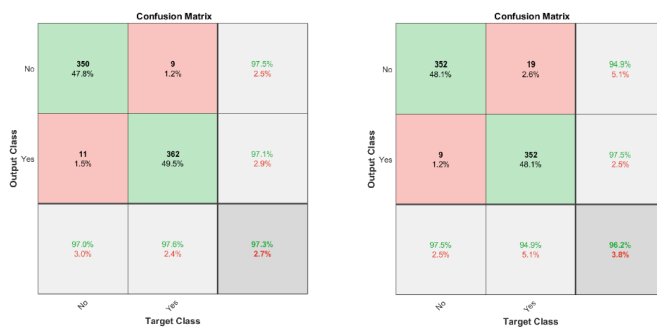


Figura 2. Matrices de confusión obtenidas para la detección de DR: (a) GoogLeNet y (b) ResNet.

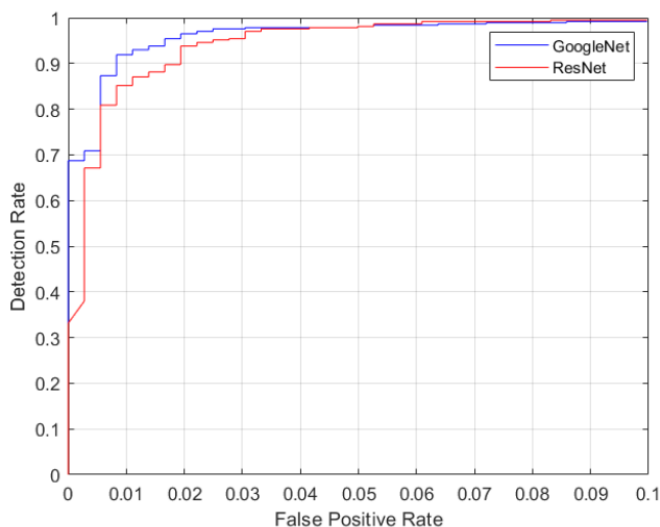


Figura 3. Curvas ROC para detección DR.

El segundo artículo seleccionado fue: **“Diabetic Retinopathy Classification Using a Modified Xception Architecture”** [7]. El enfoque propuesto en este artículo fue diseñar un framework de clasificación basados en una arquitectura Xception modificada. En el artículo eligen Xception como la red troncal (backbone network) para extracción profunda de características. En este estudio, se desarrolló un nuevo modelo de Redes Neuronales Convolucionales (CNN) basado en una versión modificada de la arquitectura Xception y con la adición

de capas profundas de CNN para la clasificación de gravedad de DR. Además se implementó el extractor profundo de Xception modificado. Se utilizó la técnica de preprocesamiento de imágenes min-pooling[8] para ayudar en la extracción de características y mejorar aún más la clasificación.

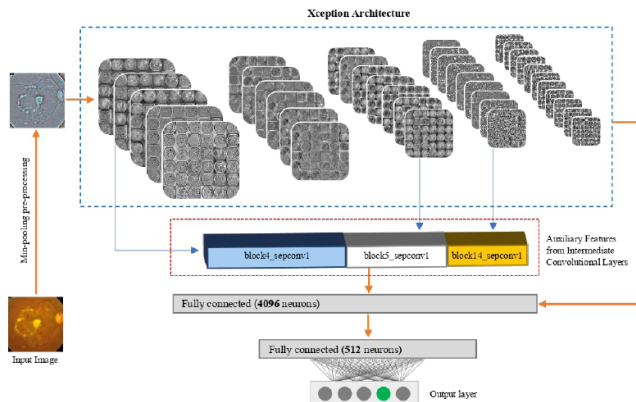


Figura 4. Ilustración de la arquitectura de red propuesta

Cuadro III

RESULTADOS DE CLASIFICACIÓN DE REDES PRE-ENTRENADAS Y ARQUITECTURA PROPUESTA. EL VALOR EN NEGRITA INDICA LO MEJOR RESULTADO; EL VALOR SUBRAYADO REPRESENTA EL SEGUNDO MEJOR RESULTADO DE LA CATEGORÍA RESPECTIVA

Metodo	Accuracy (%)	Sensitivity (%)	Specificity(%)
Xception	79.59	82.35	86.32
InceptionV3	78.72	63.64	85.37
MobileNet	79.01	76.47	84.62
ResNet50	74.64	56.52	85.71
Proposed model	83.09	88.24	87.00

El tercer artículo seleccionado: "Early Blindness Detection Based on Retinal Images Using Ensemble Learning" [9]: El proceso mediante el cual se abordó el problema completo se divide en diferentes segmentos que se realizan de forma secuencial:

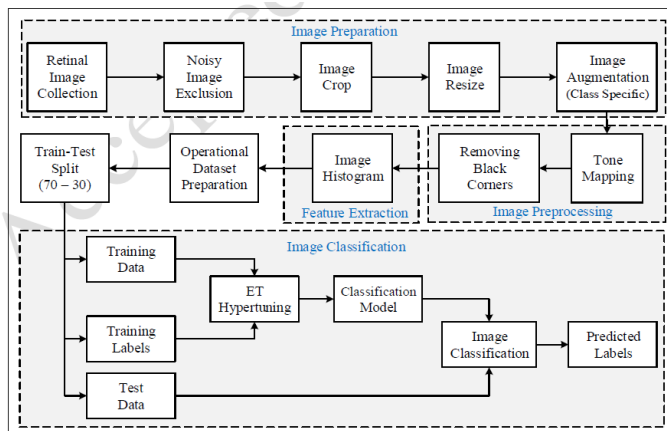


Figura 5. Metodología propuesta para la clasificación de imágenes retinianas.

Este artículo en especial no usa redes convolucionales sino que utiliza el clasificador ET optimizado que es una técnica de aprendizaje de ensamble, con éste clasificador se busca proporcionar el rendimiento máximo en el conjunto de datos preparado, utilizaron el 70 % de las imágenes para entrenar al algoritmo, y el resto 30 % de imágenes para probarlo. Realizaron Validación cruzada 10 veces para garantizar que cada muestra pertenezca al subconjunto de entrenamiento y al subconjunto de test al menos una vez. El artículo describió un método de detección de ceguera diseñado utilizando un algoritmo de ML conjunto que analiza imágenes de la retina para determinar la etiqueta de ceguera. El modelo alcanzó una precisión de clasificación del 91 % en promedio, lo que lo hace muy útil para detectar la gravedad de la ceguera. Sin embargo, el rendimiento del modelo en la identificación de algunas clases es aún mucho menor que el óptimo, lo que deja un amplio margen para trabajar con él en el futuro. A continuación los resultados:

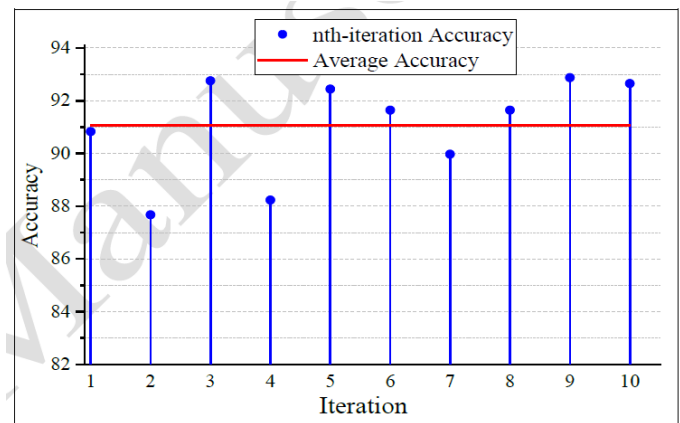


Figura 6. Precisión en diferentes iteraciones de la validación cruzada 10 veces.

True Label	No	Mi	Mo	Se	Pr	
	4133	31	113	6	19	
	31	3809	78	1	17	
	265	370	1588	113	147	
	12	5	58	2075	18	
	7	19	115	16	3104	
		Predicted Label				
		No	Mi	Mo	Se	Pr

Figura 7. Matriz de confusión de las clasificaciones.

El cuarto artículo seleccionado: "Deep Learning Approach to Diabetic Retinopathy Detection"[10]. En esta investigación, se utilizó el Puntaje ponderado cuadrático kappa de Cohen como nuestra métrica de validación. La metodología de entrenamiento y validación se realizó con versiones preprocesadas de las imágenes originales. El preprocesamiento consistió en el recorte de la imagen seguido de un cambio del tamaño de la misma. En este artículo, se propone un enfoque en la transferencia de varias etapas de aprendizaje y un método automático de detección en la etapa de retinopatía diabética para cada fotografía del fondo del ojo. Ellos usaron un conjunto de 3 arquitecturas de CNN (EfficientNet-B4, EfficientNet-B5, SE- ResNeXt50).

Model	QWK	Macro F1	Accuracy	Sensitivity	Specificity
EfficientNet-B4	0.965	0.811	0.903	0.812	0.976
EfficientNet-B5	0.963	0.815	0.907	0.807	0.977
SE-ResNeXt50 (512x512)	0.969	0.854	0.924	0.871	0.982
SE-ResNeXt50 (380x380)	0.960	0.788	0.892	0.785	0.974
Ensemble (mean)	0.968	0.840	0.921	0.8448	0.981
Ensemble (trimmed mean)	0.971	0.862	0.929	0.860	0.983
Ensemble (trimmed mean, binary classification)	0.981	0.989	0.986	0.991	0.991

Table 1: Results of experiments and metrics tracked, without using TTA.

Model	QWK	Macro F1	Accuracy	Sensitivity	Specificity
EfficientNet-B4	0.966	0.806	0.902	0.809	0.977
EfficientNet-B5	0.963	0.812	0.902	0.807	0.976
SE-ResNeXt50 (512x512)	0.971	0.853	0.928	0.868	0.983
SE-ResNeXt50 (380x380)	0.962	0.799	0.899	0.798	0.976
Ensemble (mean)	0.968	0.827	0.917	0.828	0.980
Ensemble (trimmed mean)	0.969	0.840	0.919	0.840	0.981
Ensemble (trimmed mean, binary classification)	0.986	0.993	0.993	0.993	0.993

Table 2: Results of experiments and metrics tracked, with using TTA.

Figura 8. Resultados de experimentos y métricas rastreados, con y sin usar TTA.

IV. EXPERIMENTOS

La base de datos utilizada hace parte de una competencia de Kaggle [11] que contiene aproximadamente 80GB en imágenes de diagnóstico de DR etiquetadas por médicos capacitados. Se utilizaron sólo las imágenes de entrenamiento ya que es un conjunto suficientemente grande para poder trabajar. Este conjunto con el que se trabajó contiene 35126 imágenes.

Hardware

Se contó con un portátil Intel Core i7 con 8 GB de RAM y tarjeta NVidia GeForce GT 650M tiene 1 GB de memoria que nos obligó a usar lotes muy pequeños para el preprocesamiento de las imágenes.

Preprocesamiento de imagen

En la Tabla III se puede observar que entre las clases hay un desbalance entre las clases teniendo la clase 0 (Normal) un 73.48 % del total de la base de datos que se utilizó. Por cuestiones de limitaciones en la parte del hardware y para contrarrestar este problema de desbalance se realizó un submuestreo aleatorio donde se eliminaron 21018 imágenes de la clase 0 (Normal) teniendo así 4792 imágenes de esta clase para trabajar. Así se tiene que la clase 0 es el 33.96 % de la base de datos, la clase 1 tiene el 17.31 %, la clase 2 37.51 %, la clase 3 6.18 % y la clase 4 es el 5.01 %. En total se trabajó con 14108 imágenes

A la hora de preprocesar las imágenes se tuvo en cuenta un notebook que realizaba gran parte del preprocesamiento[12]. Para poder trabajar los modelos vistos en el curso y cuyo objetivo es aplicarlos en el presente trabajo se realizó una

extracción de 150 características con la librería OpenCV en la versión 3.4.2.16, en la cual se realizó un redimensionamiento de las imágenes (512x512) y se crearon los vectores de características de cada imagen con la función detectAndCompute() de la clase Feature2D que detecta puntos clave y calcula los descriptores de las imágenes. En el punto VIII. EXTRACCIÓN DE CARACTERÍSTICAS del presente trabajo se realizó extracción de características de las imágenes con el algoritmo PCA con la diferencia que por cuestiones de capacidad de computo a la hora de correr el algoritmo de PCA se redimensionaron las imágenes a 350x350, y en lugar de 150 se hizo la extracción con 100 componentes ya que el hardware con el que se trabajó no aguantaba, y la cantidad de componentes tenía una buena varianza explicada.

A la hora de entrenar los modelos se utilizó la técnica K-Folds estratificado ya que a pesar de hacer el submuestreo los datos todavía estaban desbalanceados.

V. USO DE MODELOS DE PREDICCIÓN

Como nuestro problema es de clasificación se evalúa el uso de los siguientes modelos de predicción y se documenta los resultados de las simulaciones realizadas:

- Naive Bayes
- K vecinos mas cercanos
- Redes Neuronales Artificiales
- Random Forest
- Maquinas de Soporte Vectorial con kernel lineal y con kernel RBF.

La métrica de validación se definió como una función que toma como entrada dos vectores, $Y_{prediccion}$ y Y_{true} , y devuelve el valor correspondiente al porcentaje de las muestras mal clasificadas. El vector Y_{true} son las características verdaderas correspondientes a las entradas (o características objetivo), mientras que $Y_{prediccion}$ son las predicciones realizadas por el modelo con esas mismas entradas.

Naive Bayes

Esta sección aplica el algoritmo Naive Bayes a nuestro conjunto de datos. Para evaluar el modelo se usa K-Fold estratificado con 4 folds para distribuir las muestras de entrenamiento y de validación. Esto significa que $14108/4 = 3527$ datos estarán en cada fold. Para calcular la probabilidad o probabilidad de observar un valor real dado como X_1 utilizamos modelo de Funciones Discriminante Gaussiana con matriz de covarianza diagonal, que es equivalente al modelo clasificador Naive Bayes[13].

Finalmente los resultados al evaluar nuestros datos con Naive Bayes fueron:

- Eficiencia validación: 34.06 %
- Intervalo de confianza en eficiencia validación: 0.0052 %
- Eficiencia entrenamiento: 34.06 %
- Intervalo de confianza en eficiencia entrenamiento: 0.0017 %

Se puede observar que la eficiencia en la validación es similar a la eficiencia en el entrenamiento, esto nos indica que el modelo Naive Bayes se está sub-ajustando a los datos.

K vecinos mas cercanos

Para aplicar K vecinos más cercanos se evaluó el algoritmo con diferentes vecinos, desde 1 hasta 39 vecinos evitando usar una cantidad par de vecinos por cuestiones de sacarle el mayor provecho al algoritmo y evitando el problema de el empate.^{en} la distribución de los vecinos, es decir, que a la hora de clasificar una nueva muestra tenga igual cantidad de vecinos en cada una de las clases. Con 1 vecino el modelo arrojó el siguiente resultado:

- Eficiencia validación: 30.20 %
- Intervalo de confianza en eficiencia validación: 0.93 %
- Eficiencia entrenamiento: 100 %
- Intervalo de confianza en eficiencia entrenamiento: 0 %

Este resultado nos indica que el modelo se está sobre-ajustando a los datos. La mejor cantidad de vecinos que se encontró con este algoritmo fue de 37 vecinos, los resultados fueron los siguientes:

- Eficiencia validación: 37.71 %
- Intervalo de confianza en eficiencia validación: 0.94 %
- Eficiencia entrenamiento: 42.64 %
- Intervalo de confianza en eficiencia entrenamiento: 0.31 %

Se puede observar un sub-ajuste a los datos, pero en comparación con Naive Bayes se obtuvo un mejor resultado.

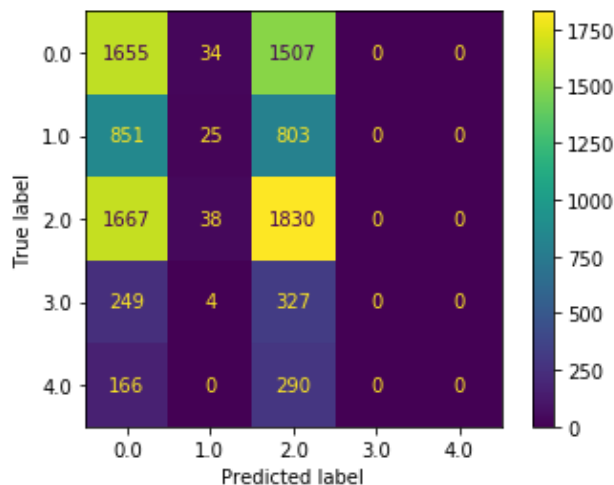


Figura 9. Matriz de confusión K vecinos

En la matriz de confusión (Figura 9) podemos observar que la mayoría de las muestras se clasificaron en la clase 0 y la clase 2, mientras que en la clase 3 y 4 no se clasificó ninguna muestra; la explicación que se le da a este resultado es por la poca cantidad de datos que hay de las clases 3 y 4, mientras que la clase 0 y 2 representan la mayor parte del conjunto de datos, es decir los datos están desbalanceados.

Redes Neuronales Artificiales

En redes neuronales artificiales se utilizó el clasificador de perceptrón multicapa, análogo a K vecinos más cercanos se buscó la mejor combinación de parámetros para entrenar la red

neural, se entrenó el modelo con la función de activación Relu, y con un máximo de 1000 iteraciones. Se hizo variación en la cantidad de capas (desde 1 capa hasta 5 capas) y en la cantidad de neuronas por capa (desde 10 hasta 120 neuronas por capa en múltiplos de 10). En total se corrieron 55 experimentos con los diferentes parámetros, donde los mejores resultados en cuanto a la eficiencia en la validación se obtienen de al rededor el 35 %.

Con 2 capas y 60 neuronas en cada capa se obtiene una eficiencia en la validación del 35.24 % y una eficiencia en la entrenamiento del 96.61 % lo que nos lleva a concluir que hay un sobre-ajuste en el modelo, análogamente con 3 capas y 100 neuronas en cada capa. Por otro lado con 4 capas y 10 neuronas en cada capa se obtiene una eficiencia en la validación es del 35.64 % y la eficiencia en el entrenamiento es de 51.81 %; aquí podemos hablar de un sub-ajuste del modelo a los datos de entrenamiento. Análogamente con 5 capas y 10 neuronas también se habla de un sub-ajuste, y se puede observar un intervalo de confianza en la eficiencia de la validación del 2.93 % que nos lleva a pensar que no es tan confiable la eficiencia en la validación y por lo tanto no deben ser parámetros a tener en cuenta, a pesar de tener mejor desempeño que otros modelos con diferentes parámetros que tienen un intervalo de confianza menor.

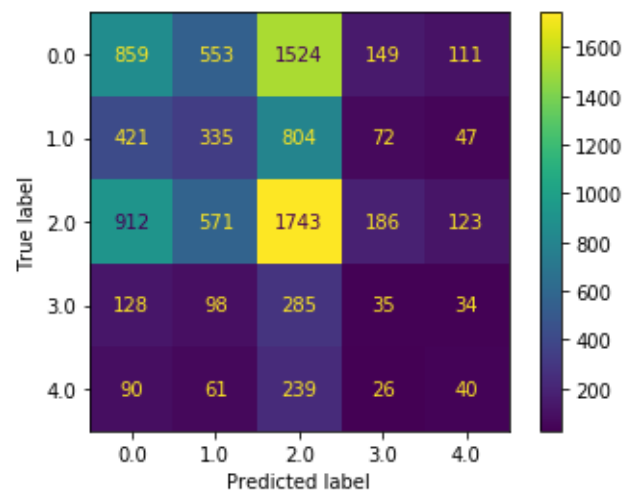


Figura 10. Matriz de confusión Redes Neuronales Artificiales

En la matriz de confusión de las redes neuronales (Figura 10) a diferencia de K vecinos, clasificó bien algunas muestras de la clase 3 y 4, pero se sigue teniendo el mismo problema de que se clasifica la mayoría de las muestras en las clases 0 y 1, por lo que se sigue notando un fuerte efecto del desbalance que hay en los datos.

Random Forest

Esta sección aplica el algoritmo Random Forest a nuestro conjunto de datos. El bosque aleatorio es un conjunto de algoritmos de árbol de decisión. Es una extensión de la agregación bootstrap (bagging) de los árboles de decisión y puede usarse para problemas de clasificación y regresión. Para

evaluar el modelo de usa K-Fold estratificado con 4 Folds. Esto significa que $14108/4 = 3527$ registros estarán en cada Fold.

Finalmente los resultados al evaluar nuestros datos con Random Forest fueron:

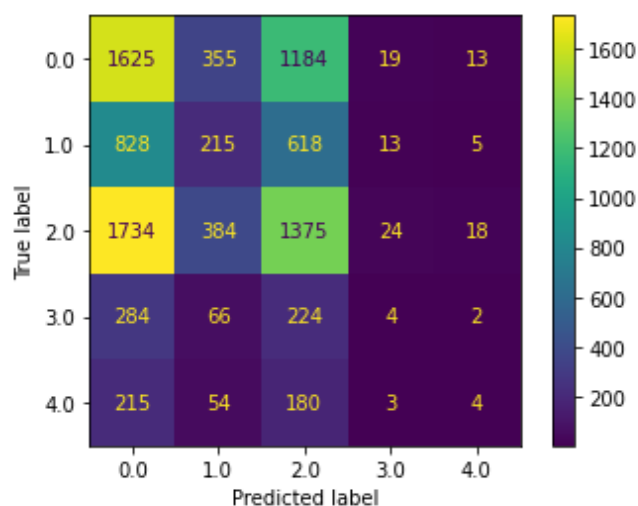


Figura 11. Matriz de confusión Random Forest

con kernel RBF.

Esta sección aplica el algoritmo SVM con kernel lineal y con kernel de Función de Base Radial (rbf) a nuestro conjunto de datos. La función para crear y entrenar el modelo usando los datos de entrenamiento es OVR (one-versus-the rest). Para evaluar el modelo se usa K-Fold estratificado con 4 folds. A continuación se pueden ver los resultados:

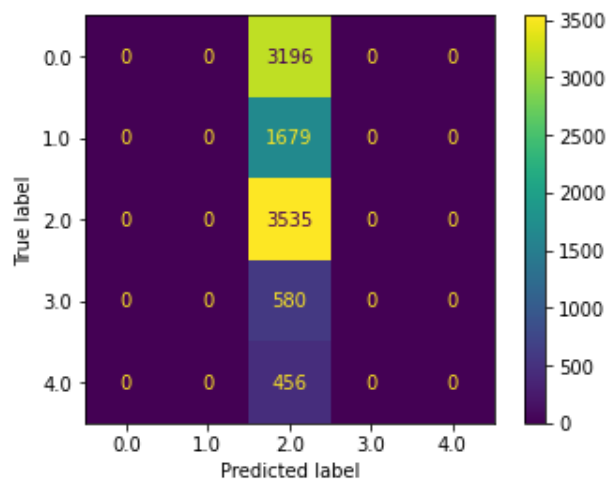


Figura 12. Matriz de confusión Random Forest

Cuadro IV
RESULTADOS RANDOM FOREST

Numero de arboles	Variables analizadas por nodo	Eficiencia en validación	Intervalo de confianza
5	5	0.3359	0.0032
	10	0.3368	0.0057
	15	0.3351	0.0046
	20	0.3320	0.0060
	25	0.3367	0.0106
10	30	0.3338	0.0048
	5	0.3554	0.0042
	10	0.3433	0.0054
	15	0.3598	0.0077
	20	0.3502	0.0064
20	25	0.3481	0.0029
	30	0.3590	0.0068
	5	0.3684	0.0012
	10	0.3604	0.0048
	15	0.3732	0.0030
50	20	0.3598	0.0028
	25	0.3675	0.0089
	30	0.3701	0.0067
	5	0.3791	0.0035
	10	0.3781	0.0042
100	15	0.3883	0.0041
	20	0.3807	0.0067
	25	0.3841	0.0099
	30	0.3847	0.0109
	5	0.3853	0.0037
	10	0.3851	0.0041
	15	0.3910	0.0046
	20	0.3884	0.0089
	25	0.3920	0.0070
	30	0.3889	0.0104

Cuadro V
SVM RESULTADO

Kernel	C	gamma	Eficiencia en validación	Intervalo de confianza	% de Vectores de Soporte
linear	0.001	0.00	0.4007	0.0067	0.9763
	0.010	0.00	0.4049	0.0065	0.9667
	0.100	0.00	0.4043	0.0075	0.9785
	1.000	0.00	0.4042	0.0069	0.9878
		0.01	0.3748	460303	0.9826
rbf		0.10	0.3748	460303	0.9982
	0.001	1.00	0.3748	460303	0.9980
		0.01	0.3748	460303	0.9933
	0.010	1.00	0.3748	460303	1.0
	1.000	0.10	0.3760	0.0010	1.0
		1.00	0.3748	460303	1.0
		0.01	0.3707	0.0024	0.9966
	10.000	0.10	0.3764	0.0008	1.0
	100.000	NaN	0.0501	616291	0.0

VI. ANÁLISIS INDIVIDUAL DE CARACTERÍSTICAS

Se realizó un análisis individual de las características de acuerdo con el índice de Fisher, luego se seleccionaron las 100 mejores características para realizar un experimento con el modelo de SVM y el resultado fue el siguiente:

- Eficiencia validación: 37.42 %
- Eficiencia entrenamiento: 37.41 %

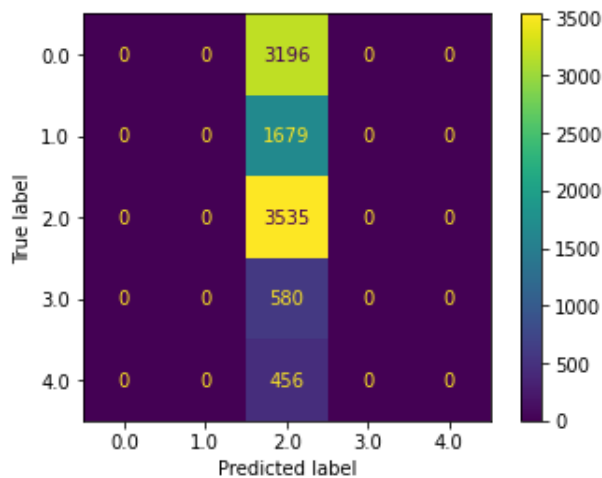


Figura 13. Matriz de confusión SVM luego de eliminar características con Fisher

Se puede observar un modelo sub-ajustado según las métricas de validación y de entrenamiento, pero en la matriz de confusión se observa que el modelo sigue clasificando todas las muestras en la clase 2.

Se hace el mismo experimento con redes neuronales artificiales y se tiene el siguiente resultado:

- Eficiencia validación: 30.42 %
- Eficiencia entrenamiento: 90.58 %

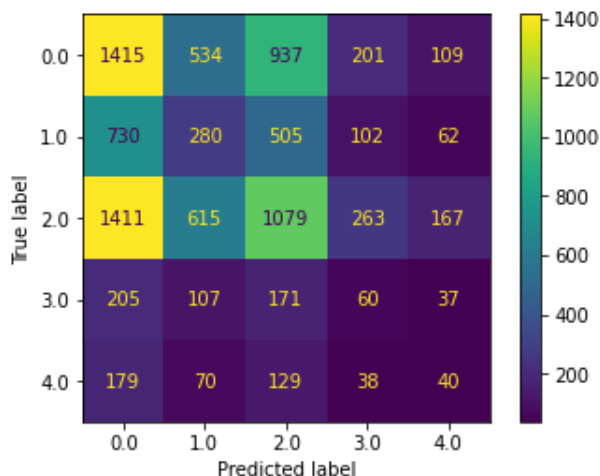


Figura 14. Matriz de confusión RNA luego de eliminar características con Fisher

Se observa que el modelo se sobre-ajusta a los datos, sin embargo es mejor clasificador que las máquinas de soporte vectorial ya las Redes Neuronales Artificiales que clasifican bien algunas muestras en las clases que les corresponde a pesar del desbalance de los datos, mientras que las máquinas de soporte vectorial clasifica todas las muestras en una sola clase. Luego de eliminar las características se noto una mejora en la clasificación de las muestras en la clase 0, 3 y 4, mientras que

en la clase 1 y 2 se nota una leve disminución en las muestras bien clasificadas.

VII. SELECCIÓN CARACTERÍSTICAS

Se realiza selección de características en el conjunto de entrenamiento Tomamos el algoritmo Fisher Score como método para explicar cómo realizar la selección de características en el conjunto de entrenamiento. Primero, calculamos los índices de Fisher de todas las características usando el conjunto de entrenamiento. Luego de ello se clasifican las características en orden descendente de acuerdo con los puntajes de Fisher se genera el índice de clasificación:

Cuadro VI
FISHER SCORE

[8 42 10 13 9 26 12 6 47 0 14 36 11 35 15 19 3 31 81 97
148 18 141 21 23 39 133 41 57 7 16 38 50 126 115 27 4
61 45 52 110 143 40 44 69 24 51 77 99 62 68 37 80 33 54
75 136 109 71 64 111 129 49 98 17 103 65 135 2 131 145
74 114 139 120 55 132 85 147 138 93 28 29 56 73 108 84
128 122 90 116 48 87 95 46 149 72 22 134 130 105 112 58
127 5 142 86 104 121 125 123 70 140 124 60 91 66 89 25
20 82 79 102 137 30 144 32 1 92 107 101 34 88 78 118 83
53 63 106 59 113 96 119 43 100 146 94 76 117 67]

Las posibles variables a usar serian las de mayor peso en el índice de Fisher.

VIII. SELECCIÓN CARACTERÍSTICAS

Se realizó selección de características con búsqueda secuencial hacia atrás (SBS). Se eligió este tipo de búsqueda ya que al tener 150 características es muy costoso computacionalmente hablando hacer búsqueda secuencial hacia adelante. Se eligieron 75 características y se evaluaron en los mejores modelos obteniendo los siguientes resultados:

- K vecinos más cercanos:
 - Eficiencia validación: 36.23 %
 - Eficiencia entrenamiento: 42.08 %

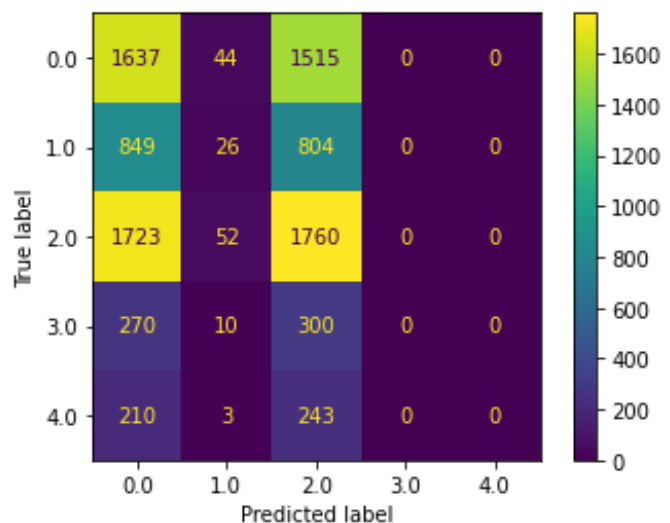


Figura 15. Matriz de confusión K vecinos luego de eliminar características con SBS

- Redes Neuronales Artificiales:
 - Eficiencia validación: 32.29 %
 - Eficiencia entrenamiento: 84.03 %

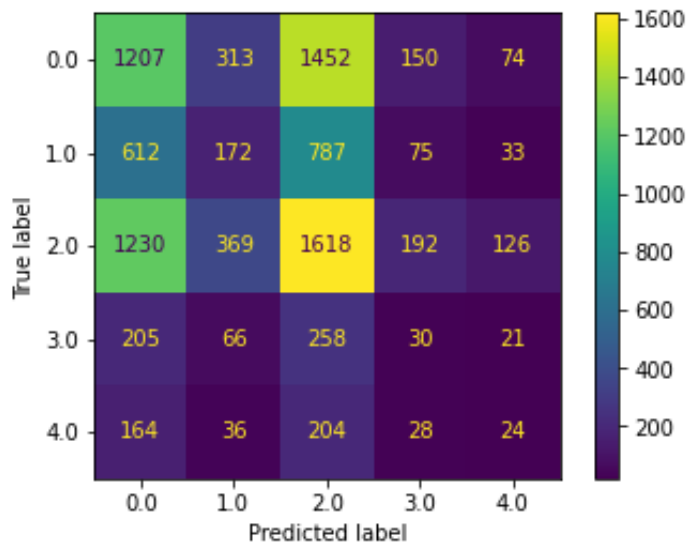


Figura 16. Matriz de confusión RNA luego de eliminar características con SBS

- Random Forest:
 - Eficiencia validación: 33.87 %
 - Eficiencia entrenamiento: 97.25 %

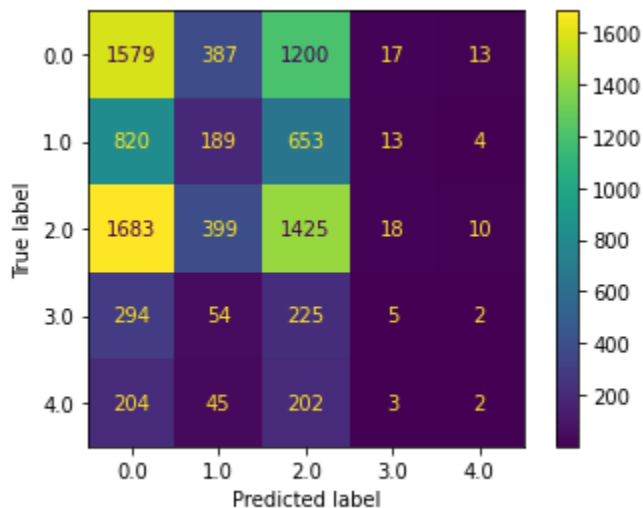


Figura 17. Matriz de confusión Random Forest luego de eliminar características con SBS

Se pueden observar resultados muy similares a los que dan los modelos al evaluarlos sin la selección de características, incluso se pueden observar mejores clasificaciones en la matriz de confusión donde se nota una mejora en cuanto a las muestras clasificadas en la clase a la que realmente pertenecen.

IX. EXTRACCIÓN DE CARACTERÍSTICAS

Para realizar extracción de características primero se realizó un preprocesamiento a las imágenes con la librería PIL donde se redimensionaron las imágenes a 350x350 con el objetivo que el computador funcionará correctamente y minimizar la posibilidad de que el algoritmo PCA se ejecutara por las limitaciones del hardware previamente descritas en la sección IV. EXPERIMENTOS.

Luego, se realizó la extracción de características ajustando el modelo con 13000 imágenes en 100 componentes que explicaban el 100 % de la varianza acumulada como se visualiza en la figura.

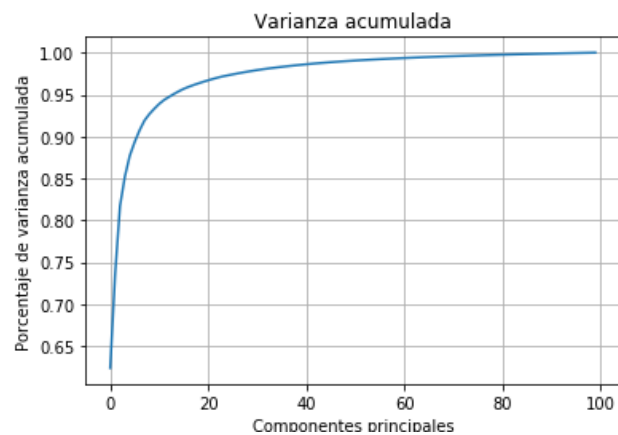


Figura 18. Varianza acumulada PCA

Evaluando los modelos con PCA se obtienen los siguientes resultados:

- K vecinos más cercanos:
 - Eficiencia validación: 38.69 %
 - Eficiencia entrenamiento: 45.67 %

En la matriz de confusión a diferencia del modelo sin PCA se observan 2 muestras clasificadas correctamente en la clase 3, pero en la clase 4 no hay ninguna muestra bien clasificada.

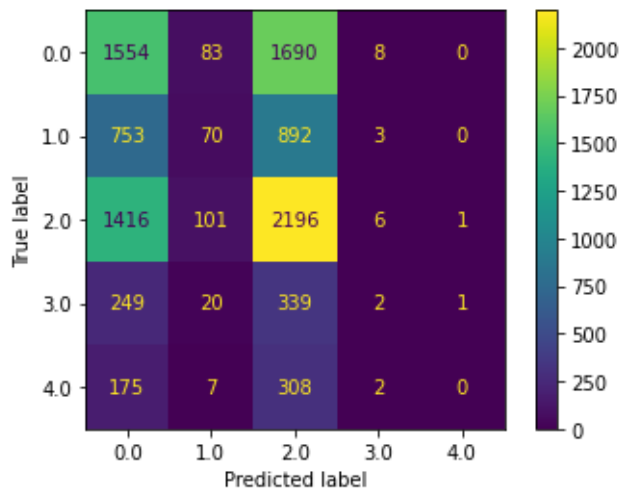


Figura 19. Matriz de confusión K vecinos PCA

■ Redes Neuronales Artificiales:

- Eficiencia validación: 31.93 %
- Eficiencia entrenamiento: 100 %

Con Redes Neuronales Artificiales usando la extracción de características con PCA, en la matriz de confusión se nota una mejor clasificación ya que en la clase 3 se clasificaron bien 43 muestras, mientras en la clase 4 se clasificaron correctamente 56 muestras. Sin embargo la eficiencia en el entrenamiento fue del 100 % lo que nos indica que existe sobre-ajuste.

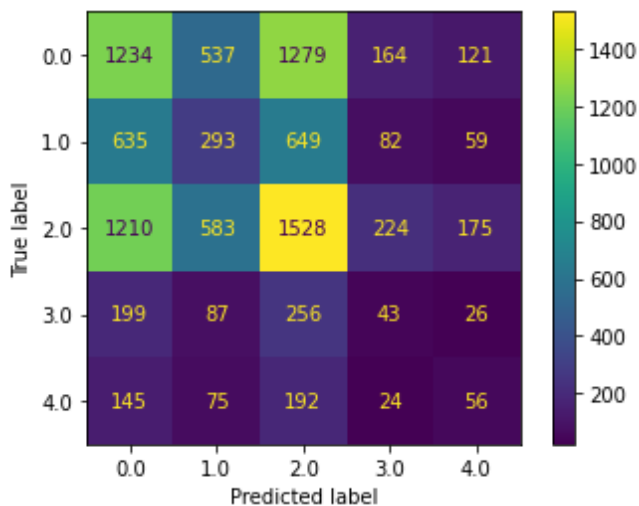


Figura 20. Matriz de confusión Redes Neuronales Artificiales PCA

■ Random Forest:

- Eficiencia validación: 34.13 %
- Eficiencia entrenamiento: 97.25 %

Ejecutando Random Forest con la extracción de características PCA se observa en la matriz de covarianza que se clasificaron bien sólo 4 muestras en la clase 3 y 2

muestras en la clase 4, además las métricas de validación nos indican que el modelo está sobre ajustándose a los datos.

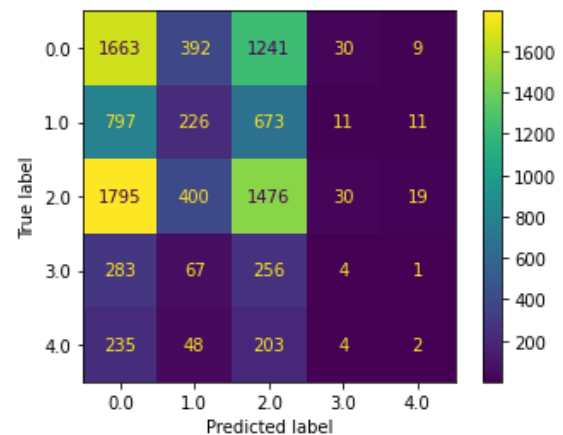


Figura 21. Matriz de confusión Random Forest PCA

■ Máquinas de soporte vectorial (SVM):

- Eficiencia validación: 39.92 %
- Eficiencia entrenamiento: 42.91 %

Con SVM se tiene el mejor resultado en cuanto a la eficiencia en la validación, la eficiencia en el entrenamiento nos quiere decir que el modelo está sub-ajustado. A pesar de tener el mejor resultado en la validación a comparación de los demás modelos, en la matriz de confusión se puede observar que las muestras solo se clasifican en la clase 0 y 2 que son las clases mayoritarias en cuanto a la cantidad de datos que poseen, todo esto debido al desbalance que hay entre las clases.

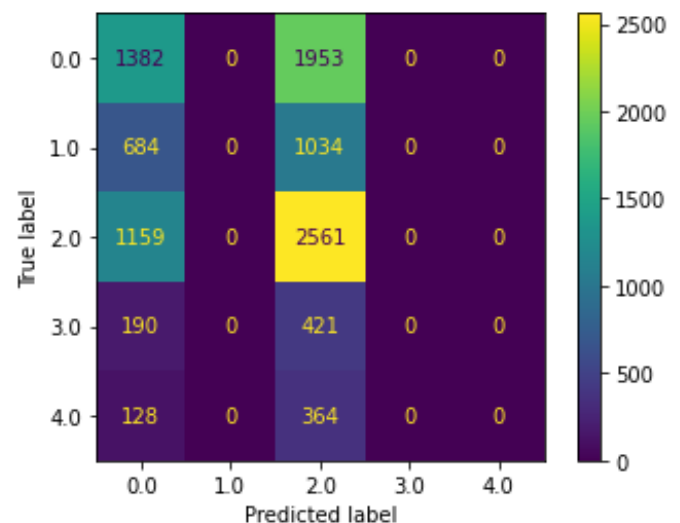


Figura 22. Matriz de confusión SVM PCA

X. DISCUSIÓN Y CONCLUSIÓN

Se puede observar que con los modelos vistos la mayoría de las veces se sub-ajusta a los datos, lo que nos quiere decir que los métodos con los que se extrajeron las características no fueron eficientes, además que el problema que se está abordando necesita modelos más complejos (computacionalmente hablando). Los papers que se plantearon como estado del arte, además de otros papers investigados usan modelos de Deep Learning donde se utilizan principalmente Redes Neuronales Convolucionales teniendo en cuenta que una de sus aplicaciones es la clasificación de imágenes. Ya que este tipo de arquitecturas no hacen parte del contenido del curso, no se incluyeron en el presente informe, pero igualmente para saciar la curiosidad de la eficiencia que tienen estos modelos tratando este problema, se corrieron varios experimentos en la nube (Ya que este tipo de arquitecturas requieren una gran capacidad de hardware) y se obtuvo una eficiencia de aproximadamente el 90 %.

Con los modelos de Machine Learning se podría mejorar la eficiencia en cuanto a problemas relacionados con la Retinopatía Diabética si se realiza un mejor preprocesamiento de las imágenes, utilizando técnicas más avanzadas en cuanto a la extracción de las características, si se tienen en cuenta variables propias del problema abordado, es decir, no extraer propiedades de las imágenes, sino propiedades del problema abordado (características que tengan en cuenta la cantidad de hemorragias, aneurismas, dilatación de los vasos sanguíneos).

REFERENCIAS

Referencias :

[1] Federación Internacional de Diabetes, «ATLAS DE LA DIABETES DE LA FID,» FID, 01 06 2019. [En línea]. Available: <https://diabetesatlas.org/es/sections/demographic-and-geographic-outline.html>. [Último acceso: 22 03 2020].

[2] World Health Organization, «Global health estimates: 2016 summary tables,» Geneva, 2016. [En línea]. Available: http://www.who.int/healthinfo/global_burden_disease/en/. [Último acceso: 22 03 2020].

[3] Kaggle, «Diabetic Retinopathy Detection,» Identify signs of diabetic retinopathy in eye images, 2015. [En línea]. Available: <https://www.kaggle.com/c/diabetic-retinopathy-detection/overview>. [Último acceso: 2020 03 20].

[4] Eyepacs, «eyepacs.com,» eyepacs, 2015. [En línea]. Available: <http://www.eyepacs.com/>. [Último acceso: 20 03 2020].

[5] Public Health England, «NHS Diabetic Eye Screening , Grading definitions for referable disease,» Enero 2017. [En línea]. Available: <https://assets.publishing.service.gov> [Último

acceso: 27 03 2020].

[6] M. S. D. S. R. C. H. N. K. K. A. Barath Narayanan Narayanana, «Understanding Deep Neural Network Predictions for Medical Imaging Applications,» Department of Electrical and Computer Engineering, University of Dayton, 1700 South Patterson Blvd., Dayton, OH, 2019.

[7] P. H. K. K. Sara Hosseinzadeh Kassani, «Diabetic Retinopathy Classification Using a Modified Xception Architecture,» University of Saskatchewan, Saskatoon, Canada, 2019.

[8] B. Graham, “Kaggle diabetic retinopathy detection competition report. 2015,” in URL <https://kaggle2.blob.core.windows.net/forum-messageattachments/88655/2795/competitionreport.pdf>.

[9] M. S. C. S. M. A.-A. N. Niloy Sikder, «Early Blindness Detection Based on Retinal Images Using Ensemble Learning,» Khulna University, Khulna, Bangladesh.

[10] Tymchenko, B., Marchenko, P., & Spodarets, D. (2020). Deep Learning Approach to Diabetic Retinopathy Detection. arXiv preprint arXiv:2003.02261.

[11] APTOS 2019 Blindness Detection — Kaggle. (2020). Retrieved 7 June 2020, from <https://www.kaggle.com/c/aptos2019-blindness-detection/data>.

[12] APTOS : Eye Preprocessing in Diabetic Retinopathy. (2020). Retrieved 7 June 2020, from <https://www.kaggle.com/rathachhat/aptos-eye-preprocessing-in-diabetic-retinopathy>

[13] Londoño Arias, J. (2020). jda-rias/ML_IntroductoryCourse. Retrieved 7 June 2020, from [https://github.com/jdarias/ML_IntroductoryCourse/blob/master/Clase %20Funciones %20discriminantes %20Gaussianas.ipynb](https://github.com/jdarias/ML_IntroductoryCourse/blob/master/Clase%20Funciones%20discriminantes%20Gaussianas.ipynb)