PROYECTO FINAL

Arias Apaza Jhon Hendrick, Medrano Callisaya Ivar Pedro
Universidad Mayor de San Andrés, La Paz, Bolivia

RESUMEN

Este proyecto presenta el uso de redes neuronales y técnicas de reducción de dimensionalidad mediante PCA para la clasificación de galaxias, estrellas y quasares utilizando un dataset del SDSS. El preprocesamiento incluyó la imputación de valores faltantes, codificación de variables categóricas y normalización. Se construyó un modelo con 100 pliegues estratificados, optimizado con Adam, logrando una precisión de 96.1% sin PCA. Sin embargo, al aplicar PCA, la mediana de la precisión disminuyó a 86.8%, lo que sugiere que la reducción de dimensionalidad, aunque mejora la eficiencia, puede implicar una ligera pérdida en el rendimiento. Este estudio destaca la efectividad de las redes neuronales para clasificación astronómica y muestra cómo la reducción de dimensionalidad puede mejorar la eficiencia sin comprometer excesivamente la precisión.

Palabras clave: Clasificación, PCA, Redes Neuronales, Galaxias, Quasares, Estrellas, SDSS, Precisión.

ABSTRACT

This project presents the use of neural networks and dimensionality reduction techniques using PCA for the classification of galaxies, stars and quasars using an SDSS dataset. Preprocessing included imputation of missing values, coding of categorical variables, and normalization. A model was built with 100 stratified folds, optimized with Adam, achieving an accuracy of 96.1% without PCA. However, when applying PCA, the median accuracy decreased to 86.8%, suggesting that dimensionality reduction, while improving efficiency, may imply a slight loss in performance. This study highlights the effectiveness of neural networks for astronomical classification and shows how dimensionality reduction can improve efficiency without excessively compromising accuracy.

Keywords: Classification, PCA, Neural Networks, Galaxies, Quasars, Stars, SDSS, Precision.

1. INTRODUCCIÓN

El análisis de datos astronómicos, como los del SDSS, es esencial para clasificar objetos como galaxias, estrellas y quasares. Este estudio compara dos enfoques de clasificación: redes neuronales y el uso de PCA para la reducción de dimensionalidad. Se evalúan las diferencias en rendimiento entre ambos métodos para optimizar la precisión y la eficiencia en el procesamiento de grandes volúmenes de datos astronómicos.

2. MATERIALES Y MÉTODOS

Descripción del Dataset:

El Dataset cuenta con un total de 100,000 datos extraídos del SDSS(Sloan Digital Sky Survey), todos los datos están descritos en 17 columnas y una columna que almacena las clase entre galaxias, estrellas y quasares, esta columna se identifica cómo ”y”, siendo nuestra variable objetivo para la predicción/clasificación del modelo.

Objetivo de investigación: comparar el rendimiento de redes neuronales y PCA en la clasificación de objetos astronómicos, analizando métricas como precisión, recall y F1-score, para identificar el enfoque más eficaz y eficiente para este tipo de tareas.

Proceso de Análisis

Preprocesamiento:

Primero se realizó la limpieza de datos para que estos puedan ser usados por los métodos de aprendizaje que usaremos, se identificaron y reemplazaron valores vacíos, se usó OneHot Encoding en columnas categóricas y se normalizo las columnas numéricas.

Selección del Clasificador: Se construyó un modelo de red neuronal con TensorFlow, realizando

múltiples pliegues estratificados (100 splits), optimizado mediante Adam y evaluado con matriz de confusión y reporte de clasificación.
Se aplica un modelo secuencial en el que las capas se apilan una tras otra, se tiene una base de 128 neuronas y función de activación ReLU.
Procesamiento con PCA: Para reducir la dimensionalidad del conjunto de datos, se aplicó el Análisis de Componentes Principales (PCA) tras estandarizar las variables numéricas. El número óptimo de componentes fue seleccionado según la varianza explicada acumulada, que alcanzó un 99.47%. Los datos transformados fueron utilizados para entrenar los modelos de clasificación.
Matemática detrás de PCA

PCA (Análisis de Componentes Principales) es una técnica de reducción de dimensionalidad que utiliza álgebra lineal para transformar un conjunto de datos a un nuevo espacio de características. En términos sencillos, PCA busca nuevas direcciones (llamadas *componentes principales*) que maximicen la varianza de los datos y que estén ortogonales entre sí. Aquí está el proceso de manera resumida:

Estandarización: Primero, los datos se centran (se les resta la media) y, a veces, se escalan (se dividen por la desviación estándar) para que cada variable tenga la misma importancia.

Cálculo de la matriz de covarianza: La covarianza mide la relación entre las variables. La matriz de covarianza describe cómo las variables se relacionan entre sí.

Cálculo de los vectores propios y valores propios:

Vectores propios: Son las direcciones en las que los datos varían más. Cada vector propio representa un componente principal.

Valores propios: Indican la "cantidad" de varianza explicada en esa dirección. Un valor propio grande significa que esa dirección captura una gran parte de la variabilidad de los datos.

Ordenación de los vectores propios: Los vectores propios se ordenan según su valor propio de mayor a menor, es decir, las direcciones que capturan más varianza se eligen primero.

Selección de componentes: Se seleccionan los primeros *k* vectores propios (combinados en una matriz) para reducir la dimensionalidad de los datos, manteniendo la mayor cantidad de información posible.

Proyección de los datos: Finalmente, los datos originales se proyectan sobre los nuevos vectores propios seleccionados, lo que da como resultado un conjunto de datos de menor dimensión, pero que mantiene la mayor parte de la varianza original.

3. RESULTADOS

Después de realizar la limpieza del dataset tenemos:

	alpha	delta	u	g	r	i	z	redshift	class_GALAXY	class_QSO	class_STAR
0	0.376905	0.503802	0.591347	0.558050	0.535344	0.427665	0.464377	0.090699	True	False	False
1	0.402286	0.491812	0.632603	0.584423	0.646203	0.515986	0.607035	0.111322	True	False	False
2	0.394960	0.534139	0.654888	0.576463	0.546218	0.435729	0.472194	0.092042	True	False	False
3	0.940947	0.180600	0.511384	0.629186	0.596946	0.486717	0.487460	0.133213	True	False	False
4	0.959118	0.392679	0.387463	0.335579	0.337999	0.287021	0.300043	0.016592	True	False	False

Figura 1. Columnas después de haber realizado la normalización, OneHot Encoding.
Para el PCA usamos todas las columnas del dataset original.

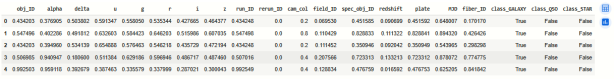


Figura 2. Columnas del dataset original a las que también se aplicó preprocesamiento.
Aplicación de la Red Neuronal
Luego de entender la matemática detrás de la implementación de nuestra red neuronal podemos usar librerías incluidas en python para hacer el código menos tedioso, el resultado es el siguiente. Se mostrarán datos del split 100

```
Matriz de Confusión en el pliegue 100:
[[9212  117   93]
 [ 145 3269    2]
 [ 221   15 2766]]
```

Figura 6. Matriz de confusión del modelo en el pliegue 100, mostrando las predicciones correctas e incorrectas para las clases galaxia, estrella y quasar.

```
Reporte de Clasificación:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97      9422
           1       0.96      0.96      0.96      3416
           2       0.97      0.92      0.94      3002

    accuracy                           0.96     15840
   macro avg       0.96      0.95      0.96     15840
weighted avg       0.96      0.96      0.96     15840
```

Figura 7. "Reporte de métricas del modelo para el pliegue 100, incluyendo precisión, recall, F1-score y soporte de cada clase, junto con los promedios macro y ponderados."

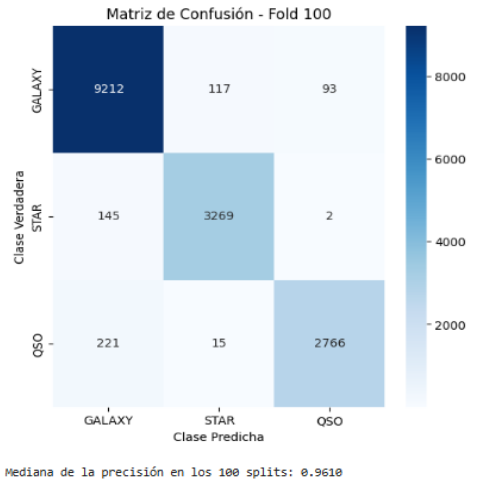
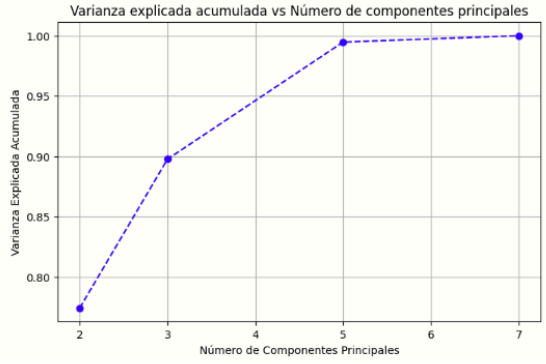


Figura 3. Visualización gráfica de la matriz de confusión del pliegue 100.

Uso de la misma red neuronal usando los datos procesados por PCA

El PCA redujo la dimensionalidad del conjunto de datos, explicando el 99.47% de la varianza con las primeras cinco componentes. El primer componente captura la variabilidad de las variables fotométricas, mientras que los siguientes combinan otras variables. Esta reducción facilita el análisis y mejora la eficiencia sin perder información relevante.




```
Varianza explicada por cada componente: [0.61175798 0.16259119 0.12337548 0.08375073 0.0132589 ]
Varianza total explicada: 0.9947342846799366
Pesos de las variables en cada componente principal:
      alpha     delta         u         g         r         i         z
0 -0.012242 -0.007146  0.386566  0.460350  0.476795  0.462908  0.443651
1  0.706566  0.707488  0.003504  0.011127  0.006658  0.006121  0.002752
2  0.704540 -0.703804  0.075571  0.018841 -0.010869 -0.028475 -0.035901
3 -0.064948  0.062561  0.749600  0.280015 -0.106172 -0.345079 -0.470327
4  0.004413 -0.012378 -0.524610  0.658632  0.294757 -0.098106 -0.440806
Datos transformados con PCA:
        PC1       PC2       PC3       PC4       PC5
0  0.902587  0.008766 -0.536321  0.814192  0.213515
1  2.980689  0.055378 -0.493175 -0.089658 -0.282682
2  1.368338  0.173486 -0.556792  1.244667  0.000230
3  1.708819  0.325914  2.042041 -0.254993  1.139533
4 -3.601340  1.073302  1.347122  0.227132  0.117141
```

Figura 4. Varianza explicada aculmulada vs numero de componentes principales y los datos transformados

```
Matriz de Confusión en el pliegue 100:
[[282   9   7]
 [ 14  71  10]
 [ 15  12  80]]

Reporte de Clasificación:
              precision    recall  f1-score   support

           0       0.91      0.95      0.93       298
           1       0.77      0.75      0.76        95
           2       0.82      0.75      0.78       107

    accuracy                           0.87       500
   macro avg       0.83      0.81      0.82       500
weighted avg       0.86      0.87      0.86       500
```

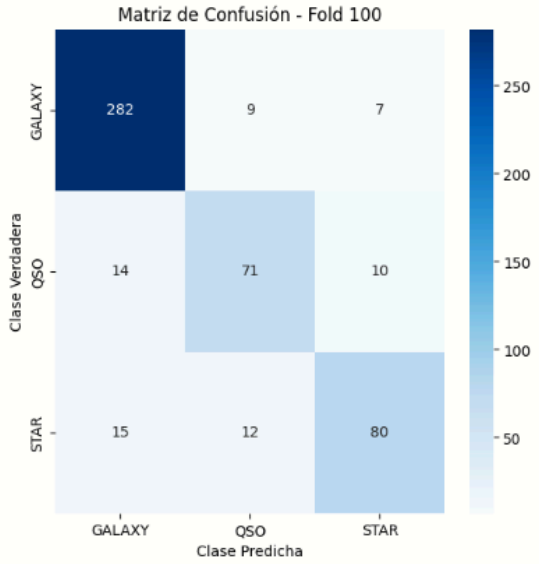


Figura 5. Resultados del entrenamiento de la red neuronal después de procesar los datos con PCA.

4.DISCUSIÓN

El modelo alcanzó una precisión del 96.1%, destacando en la clasificación de galaxias, pero con algunas confusiones entre las clases STAR y GALAXY. El uso de PCA permitió reducir la dimensionalidad del conjunto de datos, conservando el 99.47% de la varianza, lo que mejoró la eficiencia computacional y aceleró el entrenamiento del modelo. Sin embargo, la mediana de precisión en los 100 splits fue de 0.8680 con PCA, en comparación con 0.9599 sin PCA. Esto sugiere que, aunque PCA redujo el tamaño de los datos y mejoró la eficiencia, también sacrificó algo de información, lo que afectó negativamente la precisión del modelo. Las confusiones entre clases adyacentes sugieren que se pueden realizar ajustes en la red neuronal y en el preprocesamiento para mejorar la precisión. También,

los tiempos de entrenamiento podrían ser optimizados para aplicaciones en tiempo real.

5. CONCLUSIONES

El modelo propuesto logró una clasificación precisa de galaxias, estrellas y quasares, alcanzando una precisión del 96.1% sin PCA, lo que demuestra su efectividad para tareas de clasificación astronómica. La aplicación de PCA redujo la dimensionalidad del conjunto de datos, manteniendo el 99.47% de la varianza, pero resultó en una disminución en la precisión, con una mediana de 0.8680 en comparación con 0.9599 sin PCA. Esto sugiere que, aunque PCA mejora la eficiencia computacional y el entrenamiento, puede reducir ligeramente la precisión debido a la pérdida de información. Futuras investigaciones podrían enfocarse en ajustar los parámetros del modelo y explorar métodos alternativos de reducción de dimensionalidad para optimizar tanto la precisión como la eficiencia.

6. BIBLIOGRAFÍA

[1] F. Munini, D. Maggio, S. Bonometto, and M. Colavincenzo, “Dark Matter Halos in the Background Metric”, *Universe*, vol. 8, no. 2, p. 120, Feb. 2022. Disponible: https://www.mdpi.com/2218-1997/8/2/120

Recursos en línea

[2] F. Soriano. Stellar Classification Dataset, *Kaggle*, [Online].Disponible: https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17?resource=download

[3] Codificando Bits. Función de Activación, [Online]. Disponible: https://codificandobits.com/blog/funcion-de-activacion/

[4] DataCamp. Adam Optimizer Tutorial, [Online]. Disponible: https://www.datacamp.com/es/tutorial/adam-optimizer-tuto

[5] IBM, "Principal Component Analysis," [Online]. Disponible: https://www.ibm.com/es-es/topics/principal-component-analysis