Clasificación de cuentas falsas de Instagram con IA

Andrea Parra
2210062

Paula Uzcátegui
2211475

Silvia Cárdenas
2210102

Objetivo

Clasificación

Comparar diversos métodos de clasificación con IA (tanto supervisados como no supervisados) y evaluar su desempeño a la hora de identificar perfiles falsos de instagram

Dataset

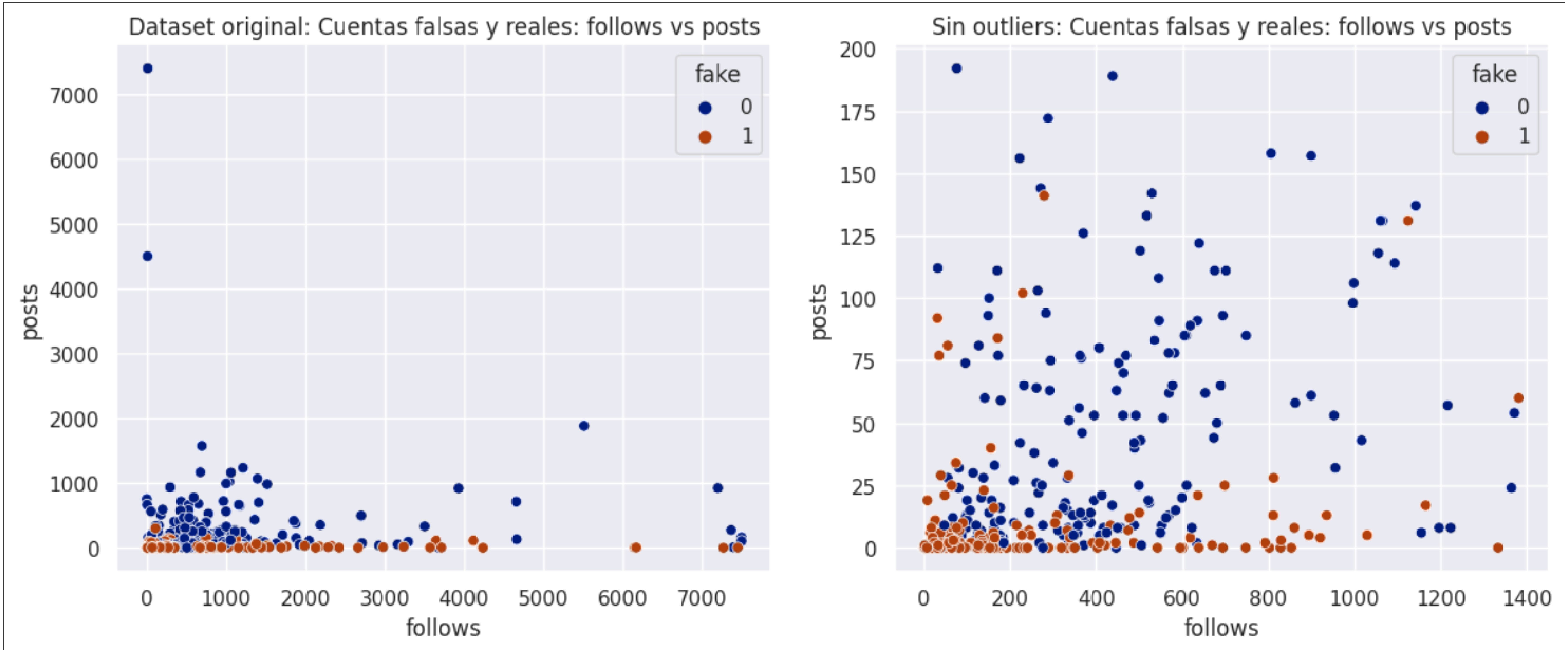
https://www.kaggle.com/datasets/free4ever1/instagram-fake-spammer-genuine-accounts

Perfiles falsos: 348, perfiles reales: 348
total: 696

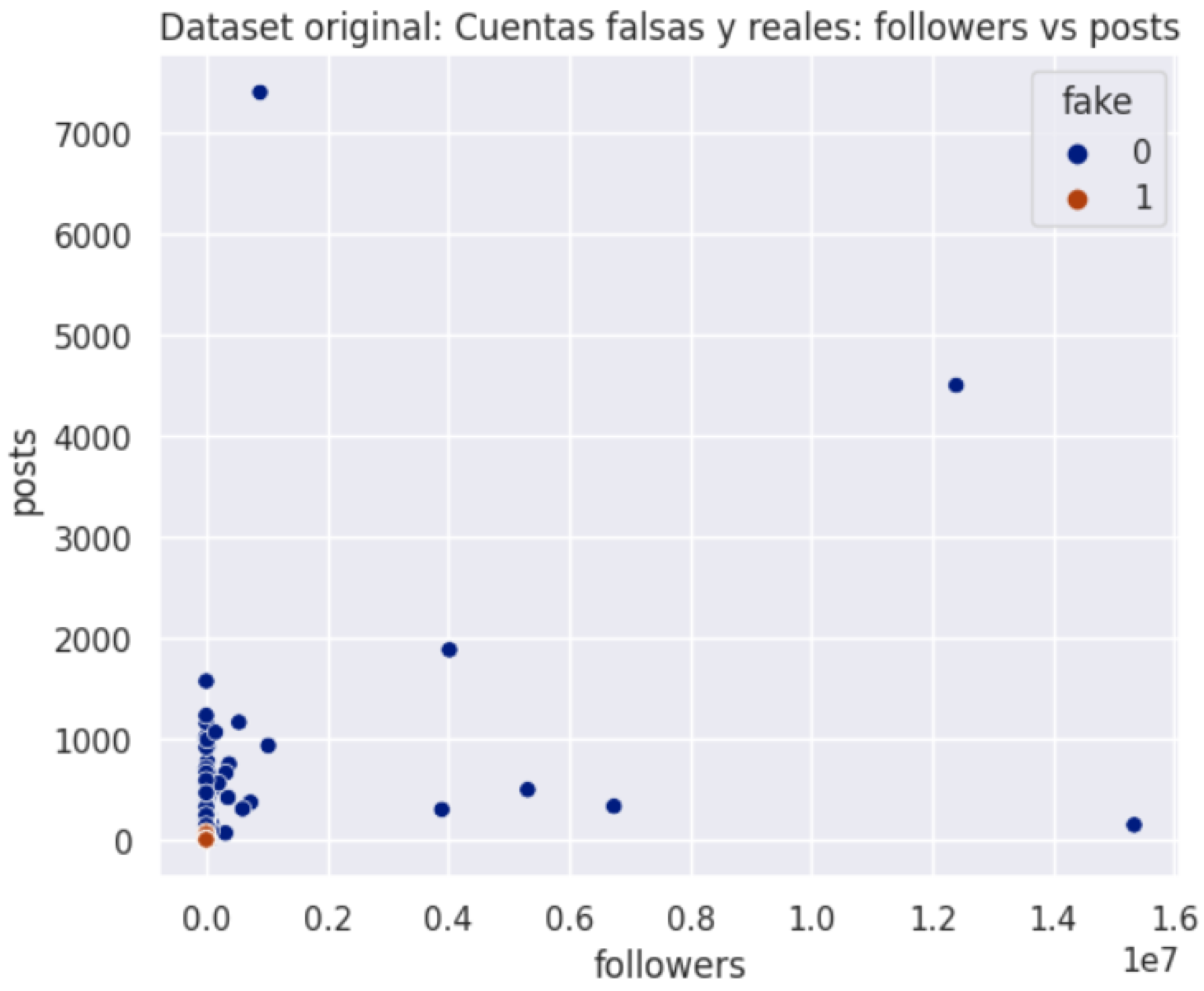
	profile_pic	num_ratio_username	fullname_words	num_ratio_fullname	name_is_username	description_length	external_url	private	posts	followers	follows	fake
0	1	0.00	2	0.0	0	126	1	0	230	2284	130	0
1	1	0.29	2	0.0	0	9	0	1	7	221	244	1
2	0	0.33	2	0.0	0	0	0	0	0	82	6	1

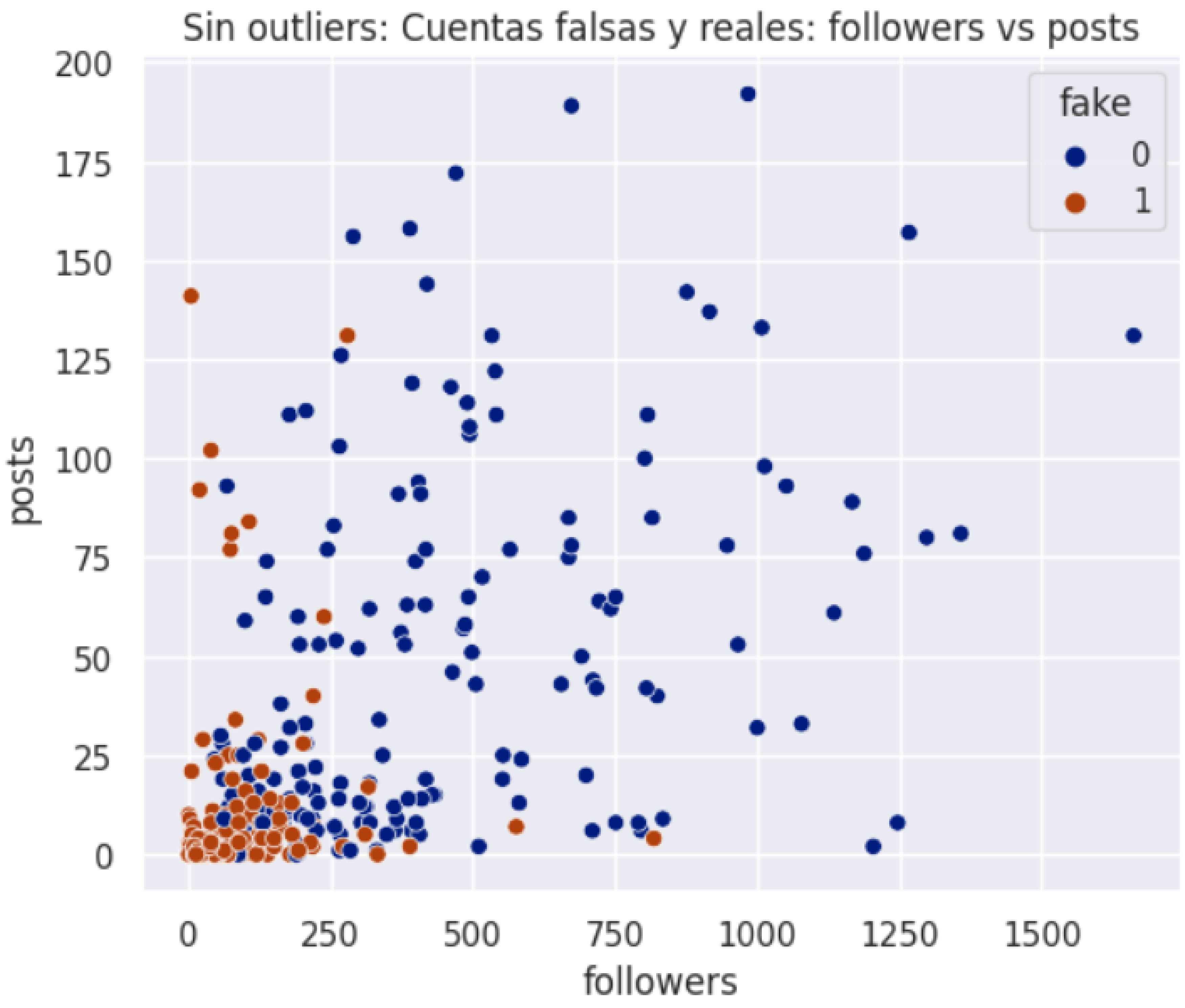
Visualización de los datos

Quitamos los outliers de todas las columnas numéricas
usando el método del indice intercuartílico



De 696 filas que tenia el dataset originalmente quedaron 428.
247 de perfiles falsos y 181 de perfiles reales
La obvia desventaja de quitar outliers es la reducción del dataset!





Métricas de evaluación

Métricas

Accuracy
Define con que frecuencia el modelo predice correctamente respecto a las observaciones totales

$$\frac{(TP + TN)}{(TP + FP + TN + FN)}$$

Precision
Mide la proporción de etiquetas predichas positivamente que son realmente correctas, detecta falsos positivos

$$\frac{TP}{TP + FP}$$

Recall
Mide la capacidad del modelo para predecir correctamente los casos positivos, detecta falsos negativos

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

Modelos supervisados

Haciendo tuning de parámetros, medimos el accuracy con cross validation

Desicion Tree

Accuracy: 90.7%
Preciscion 93%
Recall 89%

Random Forest

Accuracy: 92.8%
Preciscion 93.4%
Recall 90.8%

Gaussian Naive Bayes

Accuracy: 68%
Preciscion 61%
Recall 97%

SVM sin outliers

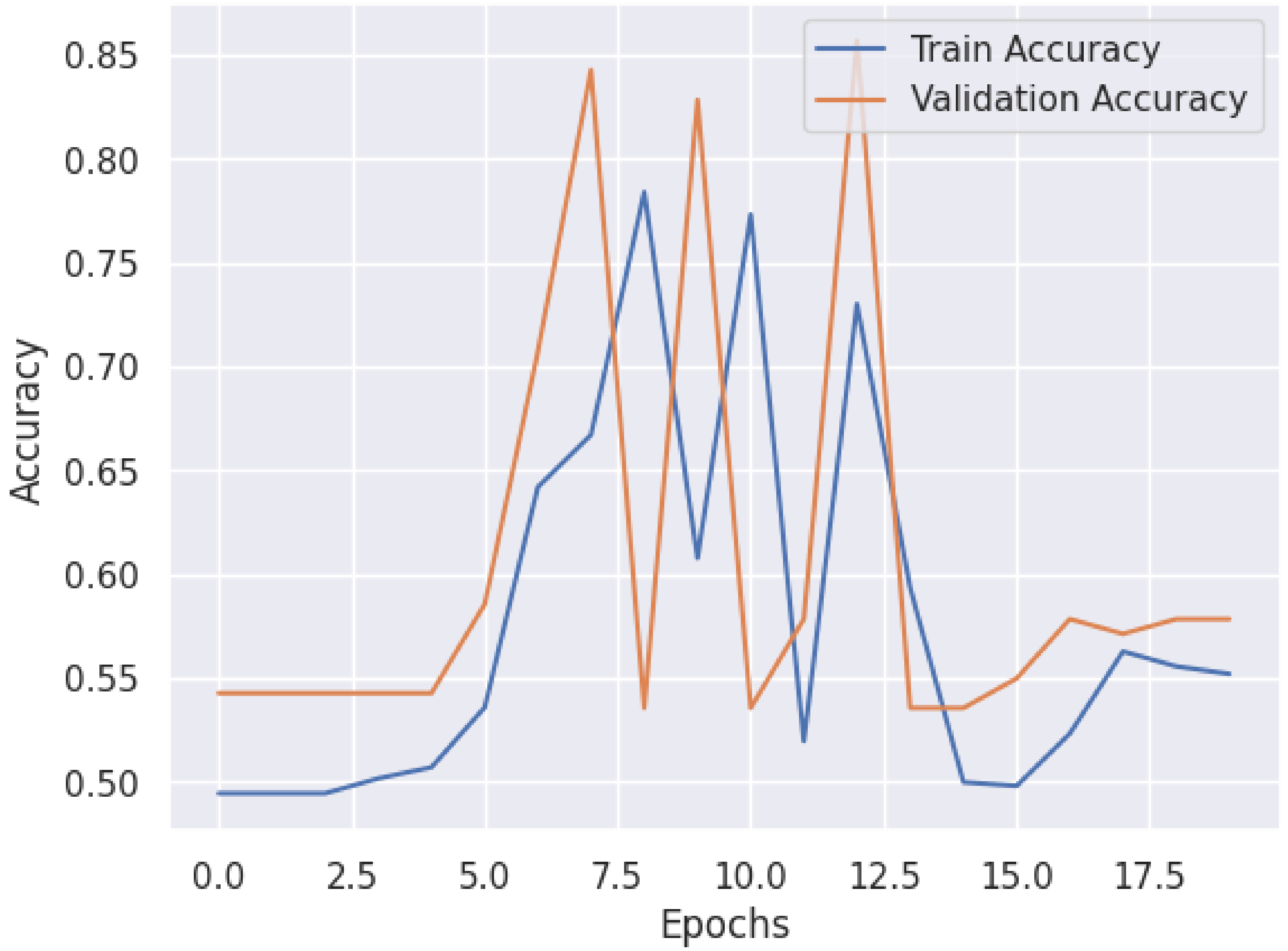
Accuracy: 89.7%
Preciscion 89%
Recall 94%

SVM con outliers

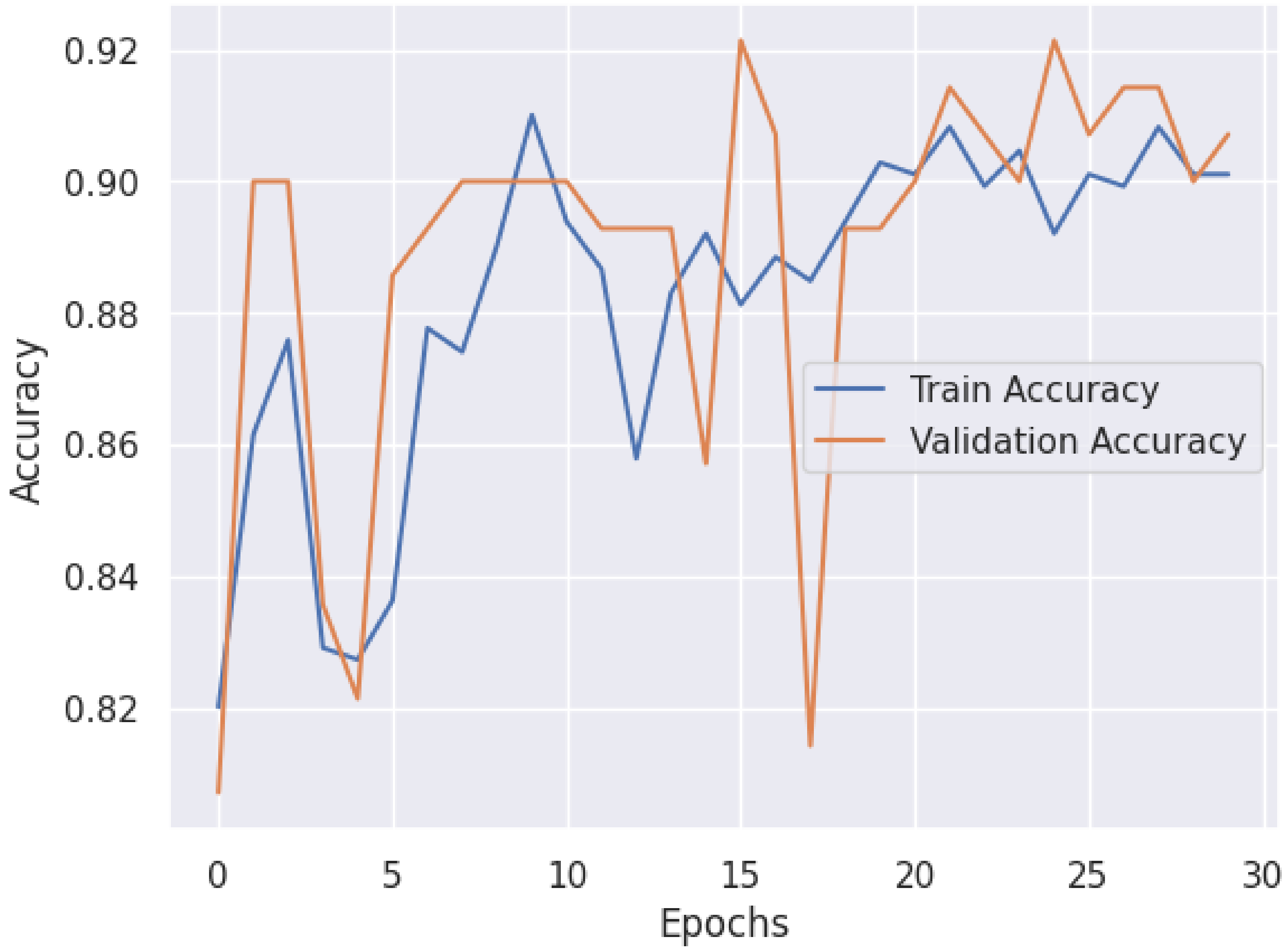
Accuracy: 59.5%
Preciscion 55%
Recall 97%

Red neuronal secuencial

Una capa oculta.
Epochs = 20
Batch size = 32
Accuracy = 55.22%

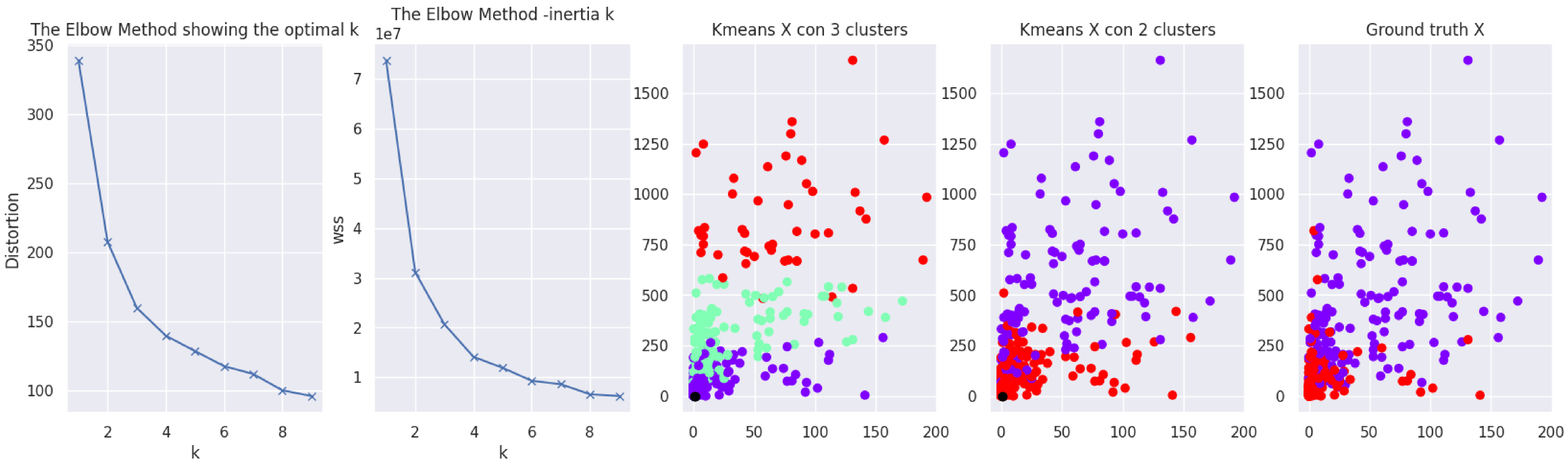


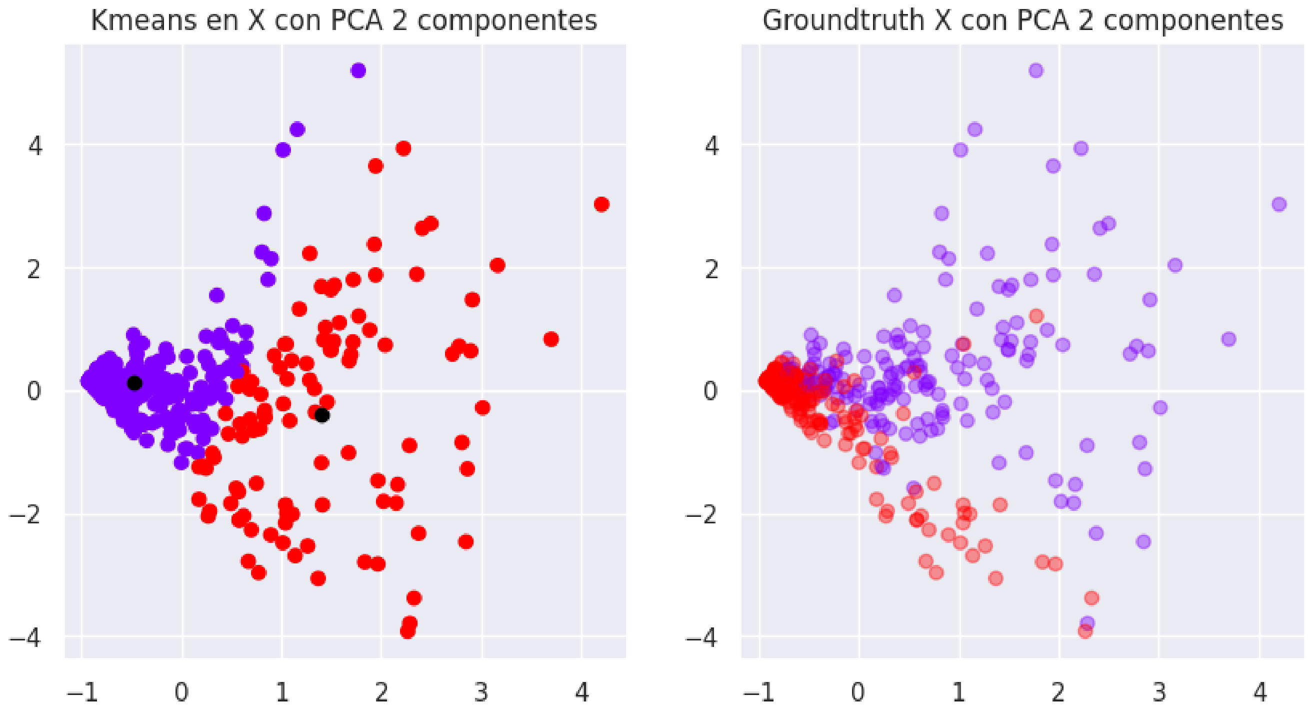
Seis capas ocultas.
Epochs = 30
Batch size = 32
Accuracy = 90.11%



Modelos no supervisados

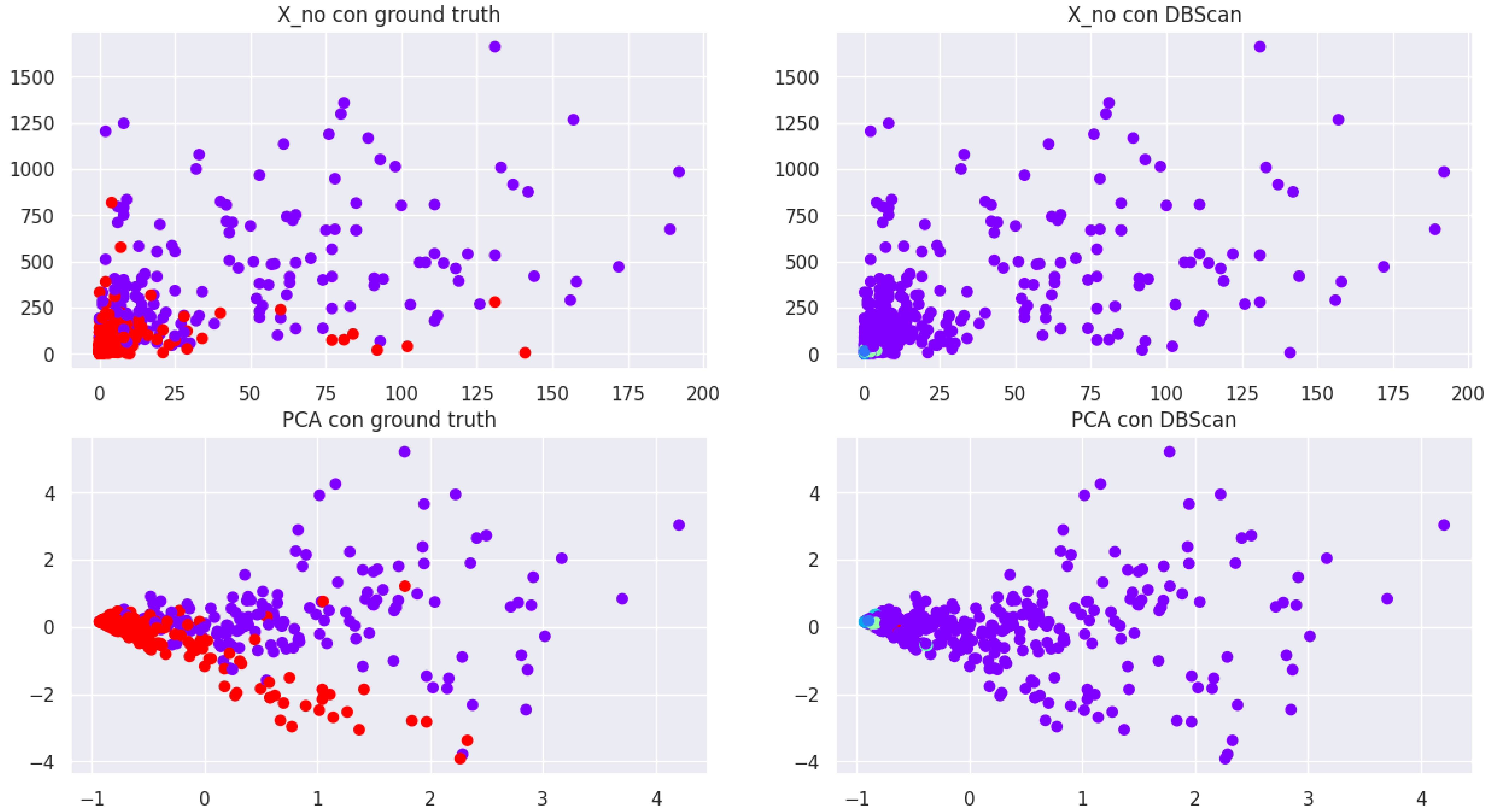
Kmeans





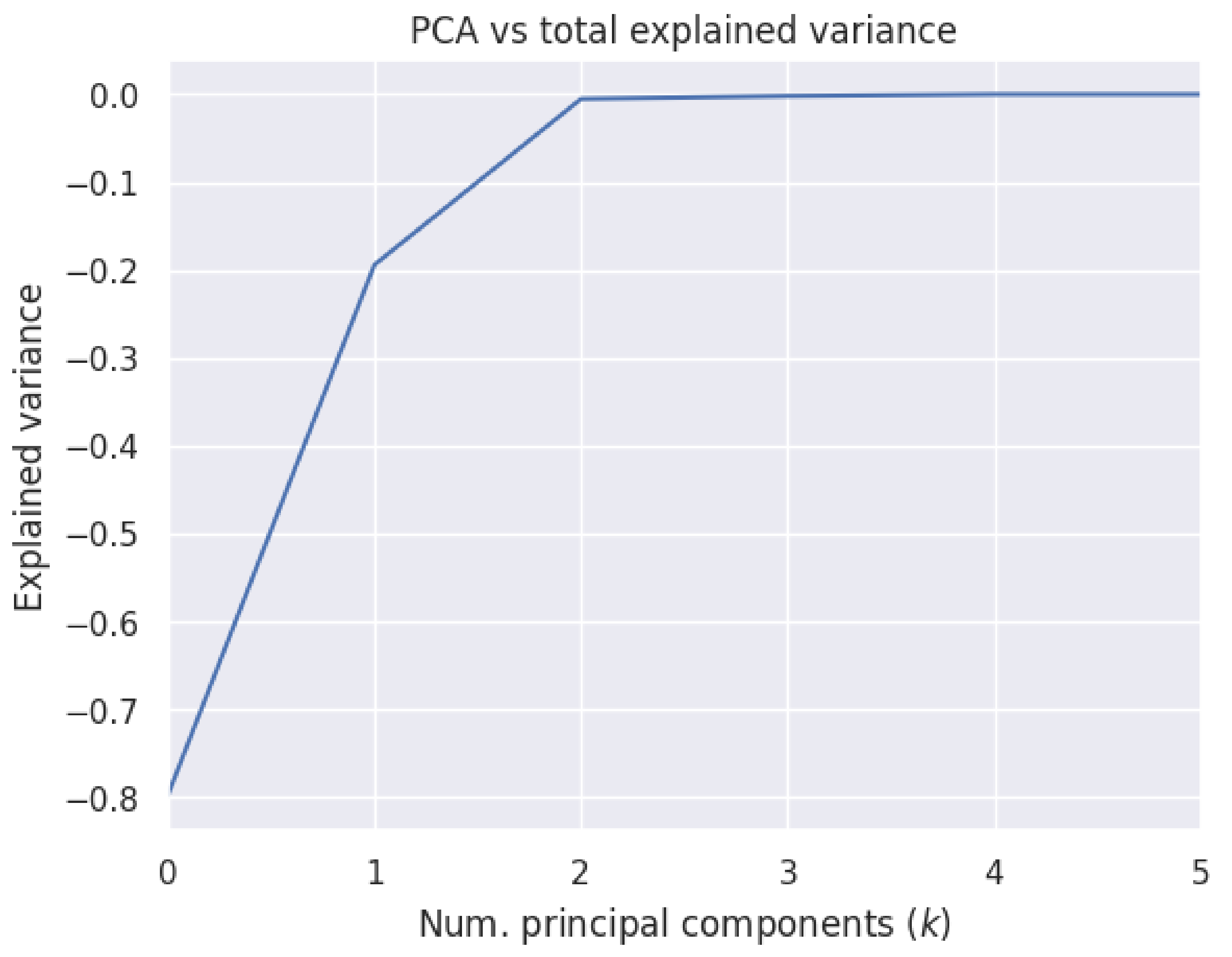

```python
distortions = []
wss = [] #Within-Cluster-Sum of Squared
K = range(1,10)
for k in K:
    kmeanModel = KMeans(n_clusters=k, n_init='auto').fit(X_no)
    kmeanModel.fit(X_no)
    distortions.append(sum(np.min(cdist(X_no, kmeanModel.cluster_centers_, 'euclidean'), axis=1)) / X_no.shape[0])
    wss.append(kmeanModel.inertia_)

plt.figure(figsize=(20,5))
plt.subplot(151)
plt.plot(K, distortions, 'bx-'); plt.xlabel('k');plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.subplot(152) #wss -> promedio distancias entre clusters
plt.plot(K, wss, 'bx-'); plt.xlabel('k');plt.ylabel('wss')
plt.title('The Elbow Method -inertia k')
```

DBScan

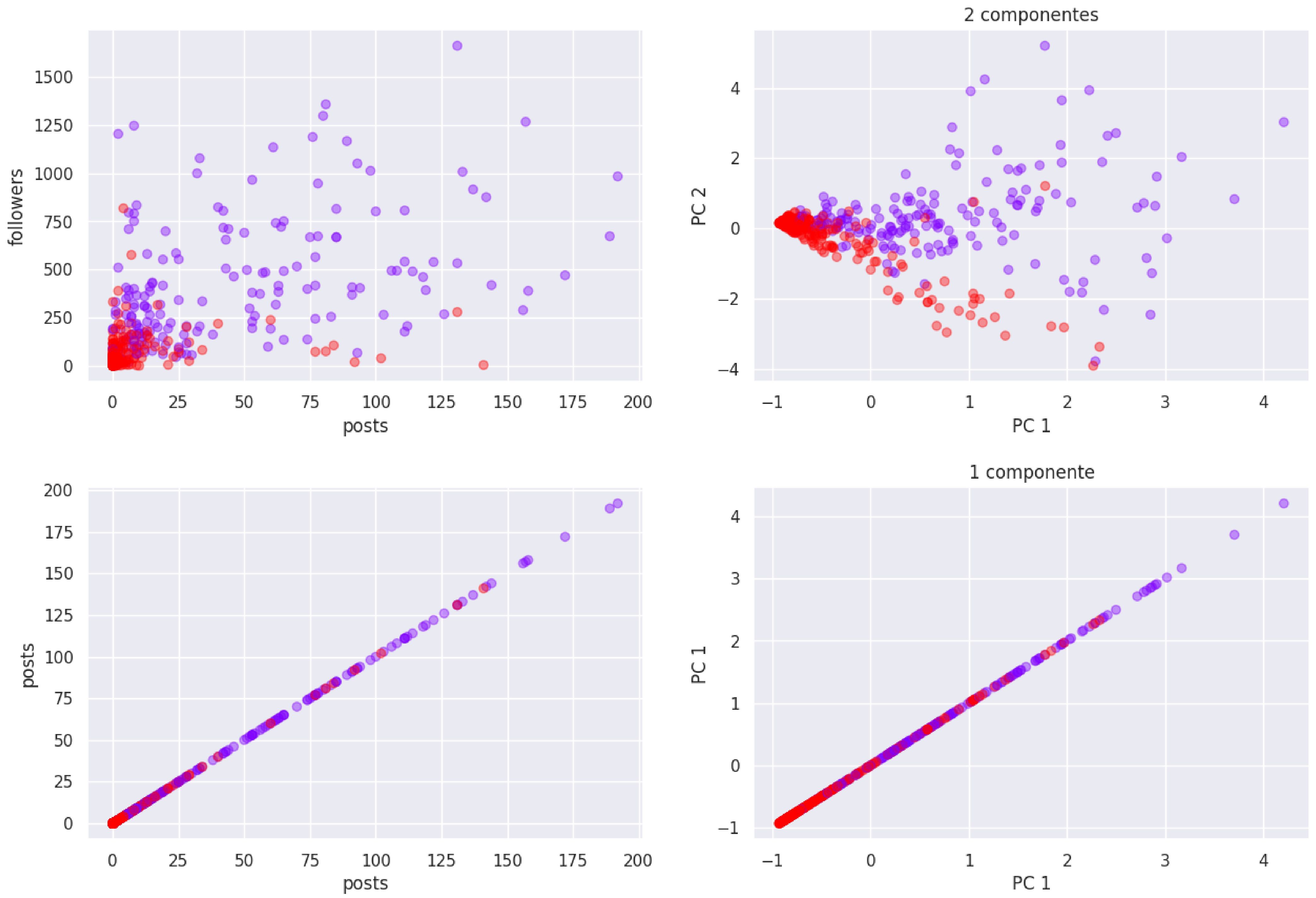

X_no con ground truth
X_no con DBScan
PCA con ground truth
PCA con DBScan

PCA

Se trabajó sin outliers




```
([0,
  0.7986369697720469,
  0.9929232912714748,
  0.9981378774449019,
  0.9999942526003001],
 array([7.98636970e-01, 1.94286321e-01, 5.21458617e-03, 1.85637516e-03,
        2.90964943e-06, 1.46999566e-06, 8.84261059e-07, 2.13277236e-07,
        1.83090540e-07, 8.71257815e-08, 3.89513256e-37]))
```



La explicativa acumulada con 2 componentes fue: 0.992923
con 1 componente: 0.798636

- La tarea de clasificar perfiles falsos se puede realizar con modelos de inteligencia artificial obteniendo buenas predicciones.
- El mejor modelo fue RandomForest con 92% accuracy, seguido por DecisionTree con 90% y la red neuronal con 90%