Práctica Sistema de Recomendación

Adrián López Gude

Ejercicio 1: Exploración inicial.

- Dataset shape : (100836, 4)
- Duplicated values : 0
- Num users : 610
- Num ratings : 10
- Num products : 9724

```
def summarize_dataset_info(dataset):

    users = dataset['userId']
    rating = dataset['rating']
    products = dataset['movieId']

    missing_values = dataset.isna().sum()
    duplicates = dataset.duplicated().sum()

    print("\nMising values :")
    print(missing_values)
    print("\nDuplicated values :", duplicates)

    print('Num users : ', users.nunique())
    print('Num ratings : ',rating.nunique())
    print('Num products : ',products.nunique())

    print("Dataset shape : ", dataset.shape)
```

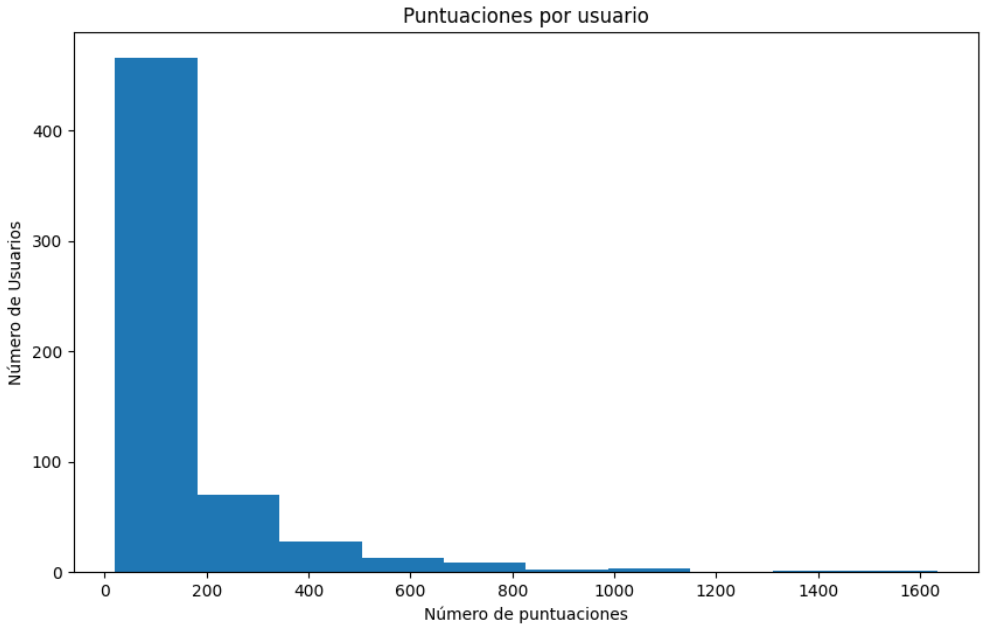
Ejercicio 2 : Limpieza del dataset

- Dataset shape : (80828, 4)
- Duplicated values : 0
- Num users : 592
- Num ratings : 10
- Num products : 2269

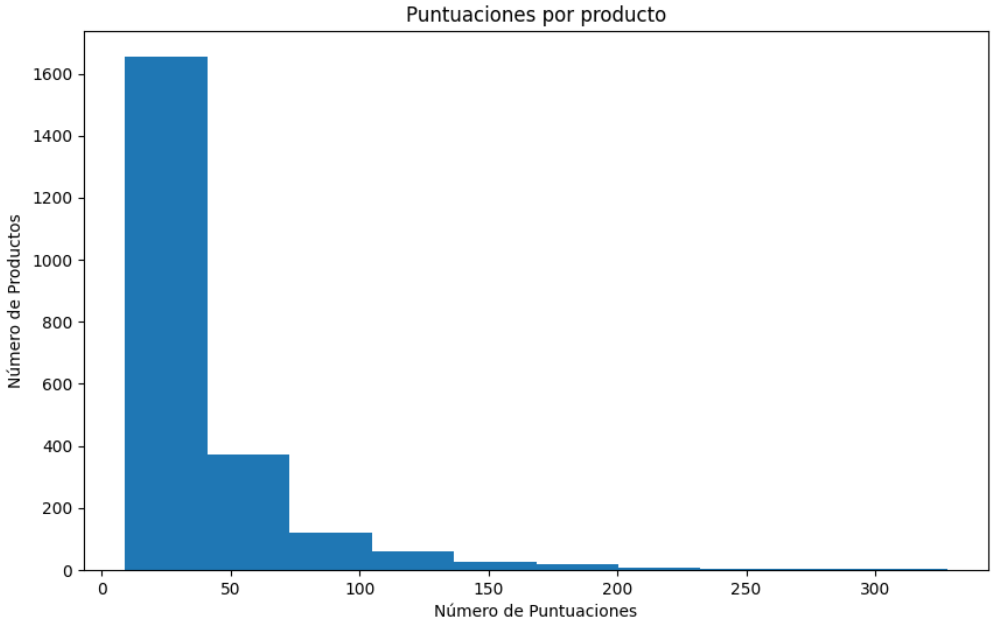
```
def clean_dataset (dataset) :

    product_counts = dataset['movieId'].value_counts()
    cleaned_df = dataset[dataset['movieId'].isin(product_counts[product_c

    user_counts = cleaned_df['userId'].value_counts()
    cleaned_df = cleaned_df[cleaned_df['userId'].isin(user_counts[user_co

    return cleaned_df
```

Ejercicio 3.

- Histograma de puntuaciones por usuario

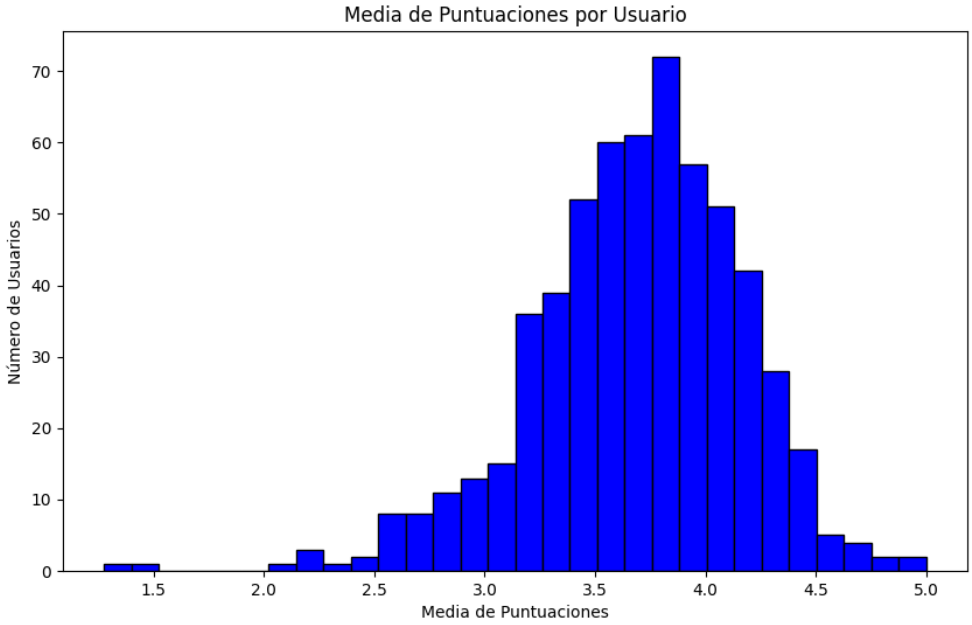


- Histograma de puntuaciones por producto

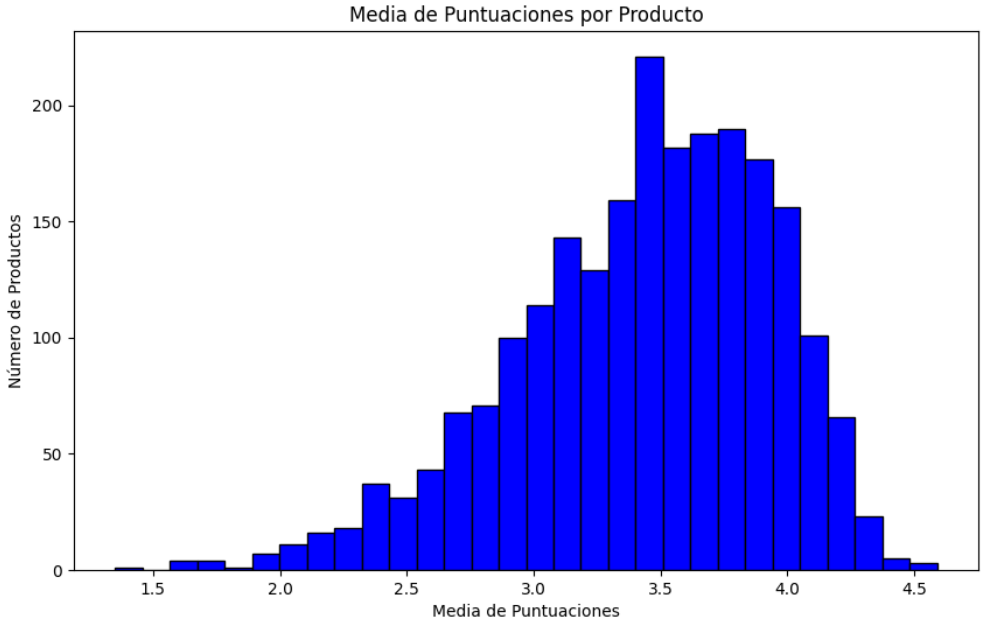


Ejercicio 4.

- histograma de la media de puntuaciones por usuario

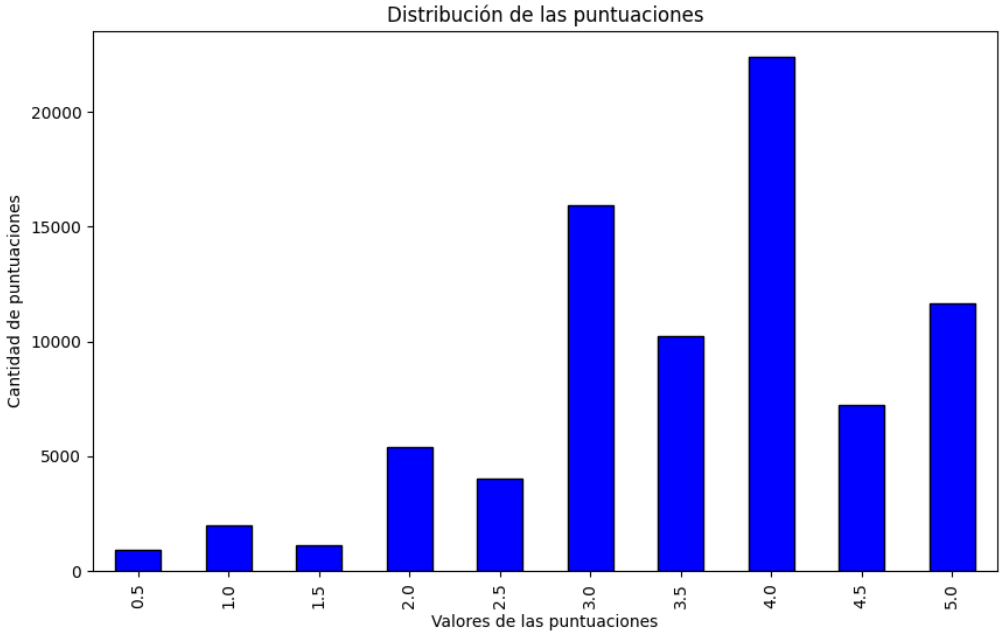


- histograma de la media de puntuaciones por producto



Ejercicio 5.

- Diagrama de distribución



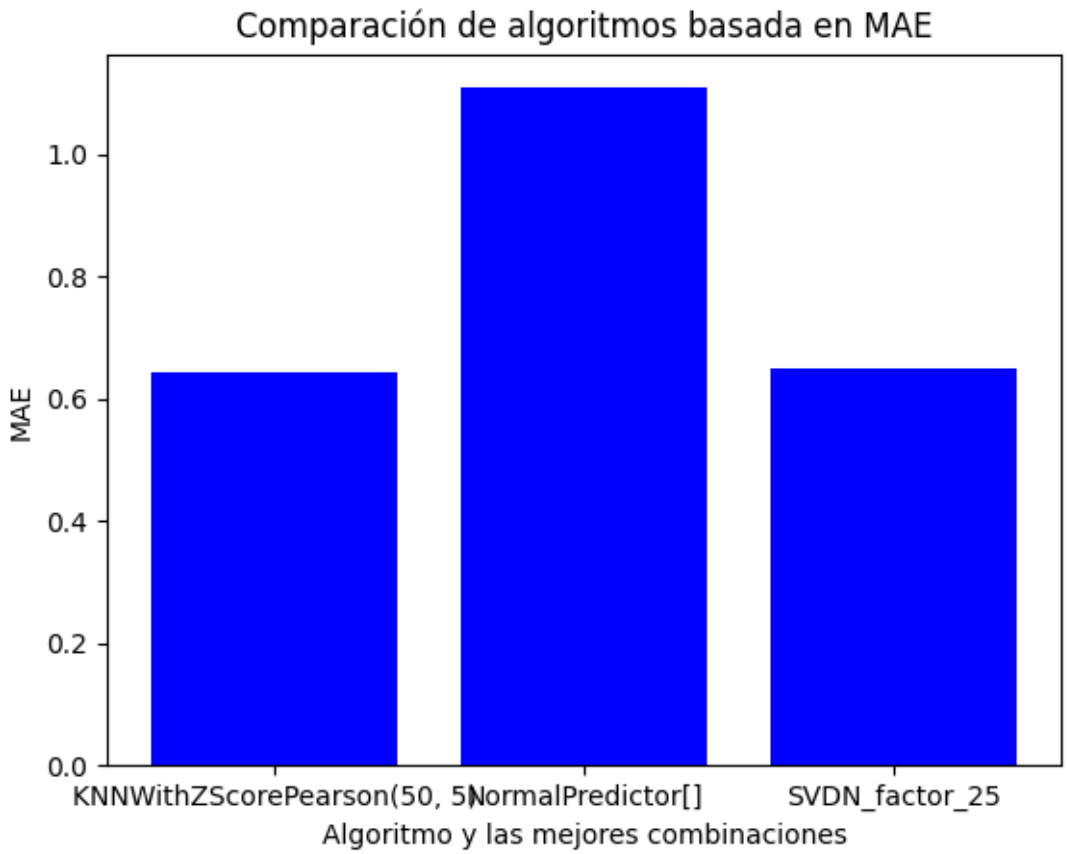
Ejercicio 6.

```
def create_surprise_object (dataset):

    min_rating = dataset['rating'].min()
    max_rating = dataset['rating'].max()
    print(min_rating)

    reader = Reader(rating_scale=(min_rating,max_rating))
    data = Dataset.load_from_df(dataset[['userId', 'movieId', 'rating']]

    print('El tipo del dataset es ', type(data))

    # Setting seed to make code reproducible
    my_seed = 311
    random.seed(my_seed)
    np.random.seed(my_seed)

    folds = set_my_folds(data)
    return data,folds
```

Ejercicio 7 y 8.

Para establecer la métrica MAE global de cada algoritmo, se ha hecho una media con los valores mínimos de lo MAEs de cada fold. Para establecer los mejores parámetros del knn se ha hecho una votación por mayoría con los parámetros que nos daba un mínimo MAE por fold. Se obtuvieron los siguientes resultados.

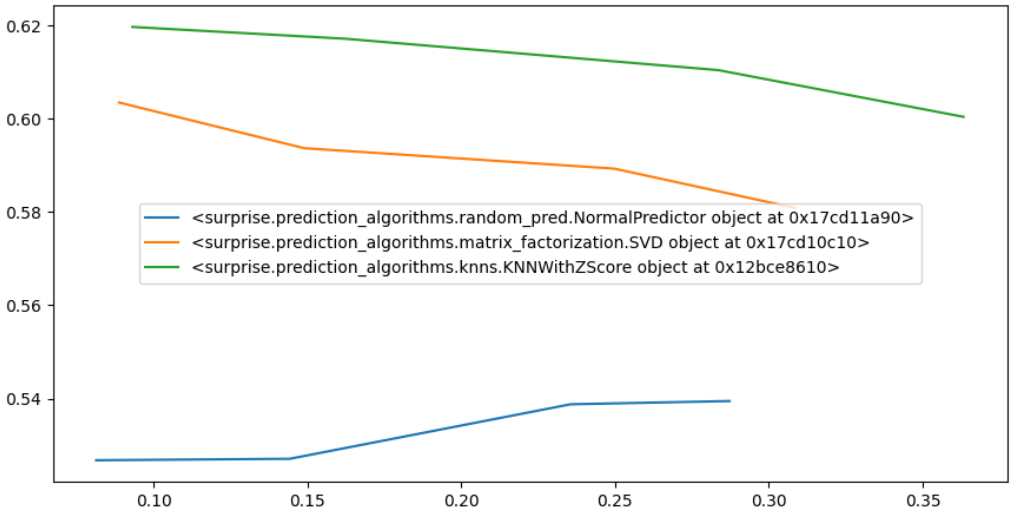
Algoritmo	Best params	MAE (global)	Standard Deviation
KNN	(k=50, min_k=5)	0.64218	0.00387
SVD	n_factors=25	0.64825	0.00229
NormalPredictor	None	1.1084	0.00557



Como conclusión, si nos basamos en la métrica MAE, podemos identificar que KNN es el mejor algoritmo ya que tiene un error menor. Cabe destacar que difiere del algoritmo SVD en las milésimas, así que podemos decir que tendrán un comportamiento muy similar. Por otra parte, si nos basamos en MAE, podemos afirmar que el algoritmo que peor se comporta es NormalPredictor ya que tiene un valor superior al resto.

Ejercicio 9.

Si nos basamos en el algoritmo que mejor precisión nos quedamos con KNN, bastante parejo con SVD. Sin embargo, podemos afirmar que Normal Predictor es peor como podríamos esperar.


0.62
0.60
0.58
0.56
0.54
0.10
0.15
0.20
0.25
0.30
0.35
<surprise.prediction_algorithms.random_pred.NormalPredictor object at 0x17cd11a90>
<surprise.prediction_algorithms.matrix_factorization.SVD object at 0x17cd10c10>
<surprise.prediction_algorithms.knns.KNNWithZScore object at 0x12bce8610>