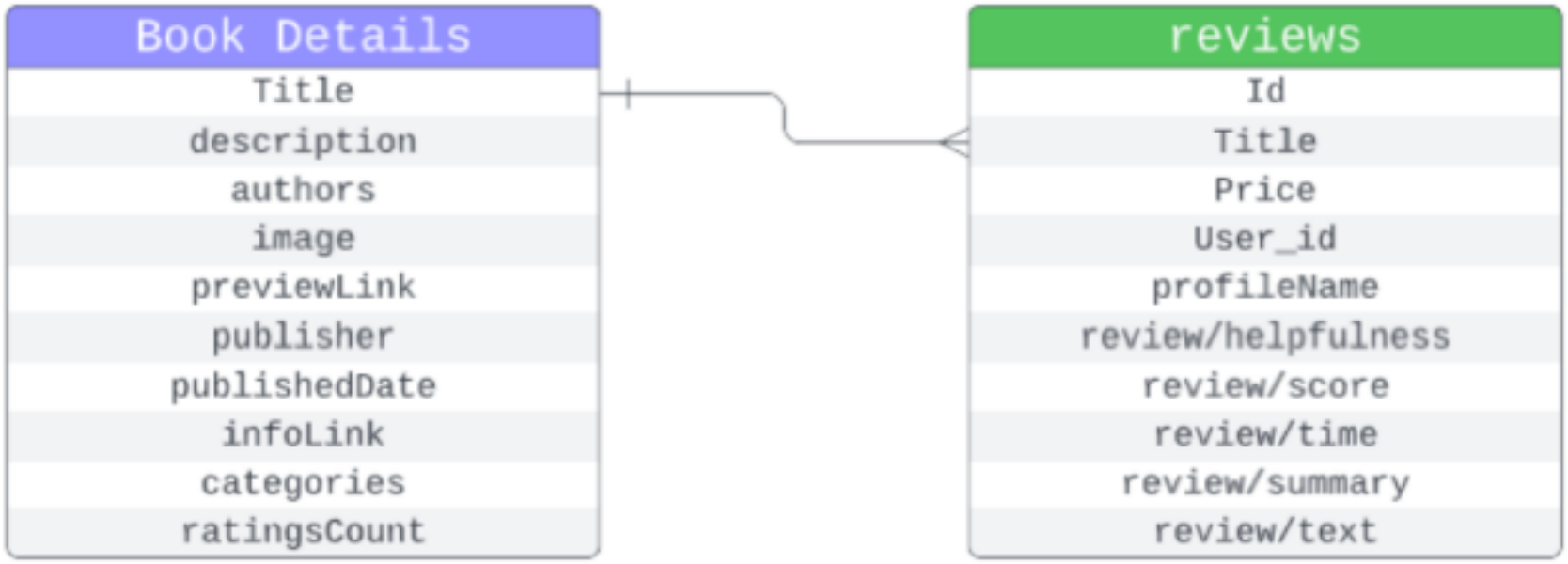
Datos Masivos

Aprendizaje No supervisado

REVIEWS EN LIBROS DE AMAZON

El datasetobtenido de kaggle contiene información de reviews de 3M de libros, este se compone de los reviews actuales de cada uno de los libros y su metadata de cada uno de los libros, dada la cantidad de información se hara uso de la Plataforma databricks la cual haremos uso de un cluster de Spark para el procesamiento de la información. El siguiente sera el esquema de las tablas actuales y su relacion.

Diagrama ER del DataSet



Introduccion

Para esta seccion se hara uso de la Plataforma Databricks para procesar la información hacienda uso de las liberias de Spark ML, para realizar un algoritmno no supervisado de clustering. Esto a fin de cada libro se pueda ir a su respective cluster basado solamente en datos categoricos y texto.

Dado que nuestro datos estan en forma de tabla solo usaremos 4 columnas de interes las cuales son: author, publisher, categories, profileName

```
1
2  df_results_snd_sampling = spark.sql('select d.Title, d.description,d.authors,d.publisher, d.categories, d.ratingsCount, r.profileName, r.Price , r.`review/summary` , r.`review/text`, r.`review/helpfulness`
   from  demo.book_reviews r inner join demo.book_details d on r.Title = d.Title where  LEN(d.categories) < 15')
```

```
1  categorical_columns = ["authors", "publisher", "categories", 'profileName']
2
3  selected_cols = categorical_columns
4  data = df_results_snd_sampling.select(selected_cols)
```

```
data: pyspark.sql.dataframe.DataFrame
  authors: string
  publisher: string
  categories: string
  profileName: string
```

```
Command took 0.10 seconds -- by jesus.ramosdv@uanl.edu.mx at 7/6/2023, 1:04:39 PM on JESUS DAVILA's Cluster
```

```
Cmd 7
```

```
1  data = data.na.fill("N/A", subset=categorical_columns)
```

```
data: pyspark.sql.dataframe.DataFrame
  authors: string
  publisher: string
  categories: string
  profileName: string
```

Aplicaremos un encoding a nuestras columnas.

```python
# Perform string indexing on categorical columns
indexers = [StringIndexer(inputCol=column, outputCol=column+"_index") for column in categorical_columns]
indexer_pipeline = Pipeline(stages=indexers)
indexed_data = indexer_pipeline.fit(data).transform(data)

# Perform one-hot encoding on indexed categorical columns
encoder = OneHotEncoder(inputCols=[column+"_index" for column in categorical_columns], outputCols=[column+"_encoded" for column in categorical_columns])
encoded_data = encoder.fit(indexed_data).transform(indexed_data)

# Combine the encoded features with the numerical columns into a single feature vector
assembler = VectorAssembler(inputCols=[column+"_encoded" for column in categorical_columns], outputCol="features")
feature_vector = assembler.transform(encoded_data)

feature_vector.show()
```

(24) Spark Jobs

```
+--------------------+--------------------+--------------+--------------------+-------------+--------------+---------------+-----------------+--------------------+--------------------+------------------+--------------------+
|             authors|           publisher|    categories|         profileName|authors_index|publisher_index|categories_index|profileName_index|     authors_encoded|   publisher_encoded| categories_encoded|    profileName_encoded|            features|
+--------------------+--------------------+--------------+--------------------+-------------+--------------+---------------+-----------------+--------------------+--------------------+------------------+--------------------+--------------------+
|     family secrets|http://books.goog...|    2013-09-01|      Noreen O'Brien|      45586.0|       12824.0|         2117.0|          72279.0|(61284,[45586],[1...|(13295,[12824],[1...|(5332,[2117],[1.0])|(454595,[72279],[...|(534506,[45586,74...|
|['Joachim E. Fest']|       Da Capo Press|   ['History']|                N/A|      53241.0|          98.0|            2.0|             0.0|(61284,[53241],[1...|  (13295,[98],[1.0])|   (5332,[2],[1.0])|  (454595,[0],[1.0])|(534506,[53241,61...|
|['Lloyd J. Harris']|                N/A|    ['Garlic']|"Amanda Burton ""...|      55067.0|           0.0|         5030.0|          82408.0|(61284,[55067],[1...|   (13295,[0],[1.0])|(5332,[5030],[1.0])|(454595,[82408],[...|(534506,[55067,61...|
|   ['Jesse Stuart']|                N/A|  ['Kentucky']|                N/A|       7462.0|           0.0|          604.0|             0.0|(61284,[7462],[1.0])|   (13295,[0],[1.0])| (5332,[604],[1.0])|  (454595,[0],[1.0])|(534506,[7462,612...|
|['Aharon Apelfeld']|David R. Godine P...|   ['Fiction']|        Eric Maroney|       6018.0|         610.0|            0.0|            255.0|(61284,[6018],[1.0])| (13295,[610],[1.0])|   (5332,[0],[1.0])|(454595,[255],[1.0])|(534506,[6018,618...|
|['Aharon Apelfeld']|David R. Godine P...|   ['Fiction']|        steven rosen|       6018.0|         610.0|            0.0|         143482.0|(61284,[6018],[1.0])| (13295,[610],[1.0])|   (5332,[0],[1.0])|(454595,[143482],...|(534506,[6018,618...|
|['Aharon Apelfeld']|David R. Godine P...|   ['Fiction']|"Mr. Rogers ""Boo...|       6018.0|         610.0|            0.0|          42134.0|(61284,[6018],[1.0])| (13295,[610],[1.0])|   (5332,[0],[1.0])|(454595,[42134],[...|(534506,[6018,618...|
|['Aharon Apelfeld']|David R. Godine P...|   ['Fiction']|                N/A|       6018.0|         610.0|            0.0|             0.0|(61284,[6018],[1.0])| (13295,[610],[1.0])|   (5332,[0],[1.0])|  (454595,[0],[1.0])|(534506,[6018,618...|
```

Command took 46.14 seconds -- by jesus.ramosdv@uanl.edu.mx at 7/6/2023, 1:04:39 PM on JESUS DAVILA's Cluster

Aplicando KMeans

Para este ejercicio se usara un numero pequeno de cluster 5, ya que algunos paginas sugieren que los generos de los Libros se pueden categorizar de una manera simple por al menos 5 generos.

```
1   # Perform clustering using K-means
2   k = 5  # Number of clusters
3   kmeans = KMeans(k=k, seed=123)
4   model = kmeans.fit(feature_vector.select('features'))
5
6
7   # Get the cluster assignments
8   predictions = model.transform(feature_vector)
9
10  # Show the clustering results
11  predictions.show()
```

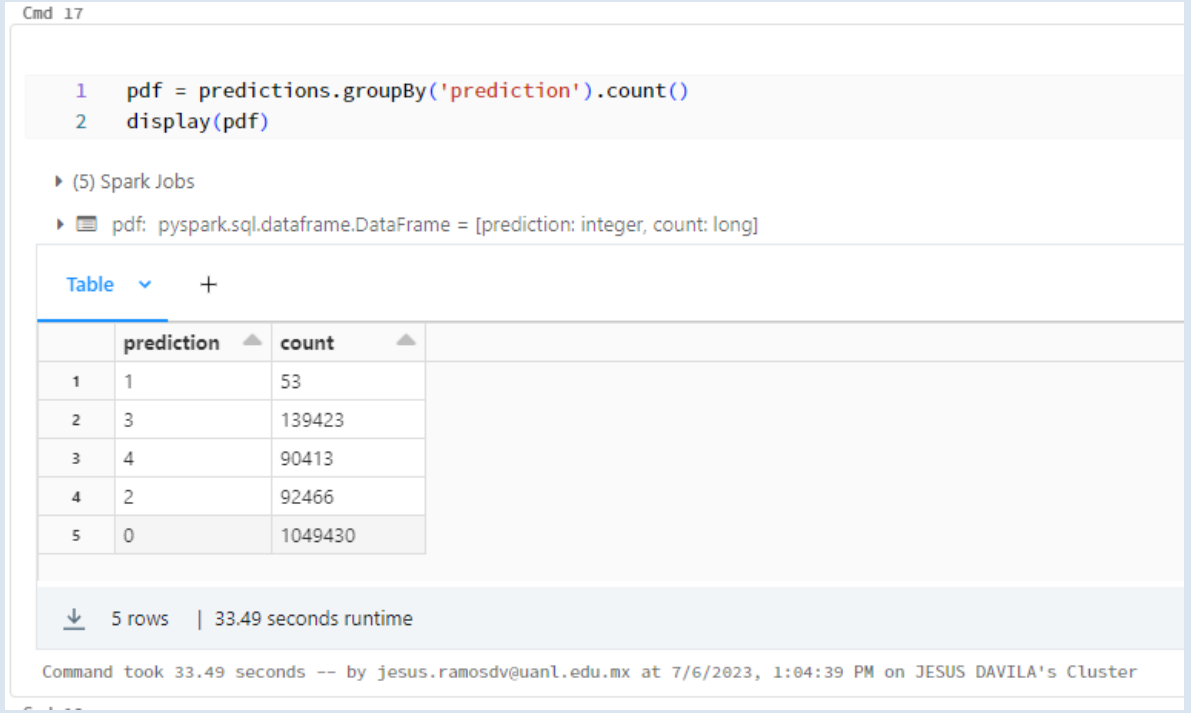
Resultados

Al observar algunos registros se muestra que tanto la categoria Fiction como Philosophy estan en clusters separados

publisher	categories	profileName	prediction
David R. Godine Publisher	['Fiction']	Lost John	0
David R. Godine Publisher	['Fiction']	Bookski	0
University of Toronto Press	['Philosophy']	"Nathan Andersen ""film lover	2
University of Toronto Press	['Philosophy']	peter simpson	2
University of Toronto Press	['Philosophy']	David Morris	2

Resultados

Se observaron 2 clusters con demasiadas observaciones, y un cluster con escasos resultados, una de los errores el cual afecto al algoritmo fueron los datos ya que hubo demasiadas observaciones con datos vacios y categorias que no eran tal cual categorias de generos de libros , lo cual afecto al modelo.


Cmd 17

```
1  pdf = predictions.groupBy('prediction').count()
2  display(pdf)
```

▸ (5) Spark Jobs

▸ pdf: pyspark.sql.dataframe.DataFrame = [prediction: integer, count: long]

Table +

	prediction	count
1	1	53
2	3	139423
3	4	90413
4	2	92466
5	0	1049430

5 rows | 33.49 seconds runtime

Command took 33.49 seconds -- by jesus.ramosdv@uanl.edu.mx at 7/6/2023, 1:04:39 PM on JESUS DAVILA's Cluster

Resumen

Se realize un aprendizaje no supervisado el cual no tuvo los resultados esperados ya que los datos tenian información incorrecta, lo cual afecto el numero de clusters y sus observaciones dentro de cada cluster, en solo algunos de los clusters si se encontraron generos que tenian similitudes pero este no fue el caso en todos los clusters, una de las partes a mejorar seria la limpieza de datos para mejorar el modelo.

Referencias

https://towardsdatascience.com/k-means-clustering-using-pyspark-on-big-data-6214beacdc8b

https://www.databricks.com/

Session 3 ML Model DM.html(html python notebook)