ADS508 Data Science_Cloud Computing

Authors: Emma Oo & Juliet Sieland-Harrris
Company Name: Etoile Cinemas Company Industry: Entertainment Company Size: Small-sized business with 25 employees

In [1]:

```
#INGESTING CRITICS DATA FILE INTO CRITICS

import boto3
import pandas as pd


s3_client = boto3.client("s3")

BUCKET='ejcinemas'
KEY='critics/critics.csv'


response = s3_client.get_object(Bucket=BUCKET, Key=KEY)
critics = pd.read_csv(response.get("Body"))
critics.head(10)
```

```
/opt/conda/lib/python3.7/site-packages/IPython/core/interact
iveshell.py:3063: DtypeWarning: Columns (3) have mixed type
s.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=res
ult)
```

Out[1]:

	Unnamed: 0	ID_Movie	Expert	Score	Review	Sentiment
0	0	7369	RogerEbert.com	88	Call Me Lucky will be an especially grueling r...	3.0
1	1	7369	New York Daily News	80	Angry, quixotic, tragic, heroic — Crimmins' li...	1.0
2	2	7369	Village Voice	80	Call Me Lucky is a loving but fair portrait of...	7.0
3	3	7369	TheWrap	75	There should be more Crimmins performance foot...	2.0
4	4	7369	Movie Nation	75	Call Me Lucky is another of those "the funnies...	3.0
5	5	7369	The A.V. Club	67	Goldthwait stays behind the camera, but his lo...	1.0
6	6	7369	Slant Magazine	63	Bobcat Goldthwait's hand too nervously tempers...	-2.0
7	7	7369	The New York Times	50	The movie strains to drum up mystery as to the...	2.0
8	8	7369	Austin Chronicle	50	You'll be the richer for spending time in Crim...	3.0
9	9	7369	Washington Post	37	Ironically, Call Me Lucky, a worshipful new do...	0.0

In [2]:

```
#INGESTING SALE DATA INTO SALE

s3_client = boto3.client("s3")

BUCKET='ejcinemas'
KEY='sale/sale.csv'


response = s3_client.get_object(Bucket=BUCKET, Key=KEY)
sale = pd.read_csv(response.get("Body"))
sale.head(10)
```

Out[2]:

	MovieName	Rank_data	PreviousWeekRank	GrossW	Theaters
0	Stuart Little	1	1	13012299	2979
1	The Green Mile	2	3	12521303	2678
2	The Talented Mr. Ripley	3	2	11780319	2316
3	Any Given Sunday	4	4	10971011	2505
4	Galaxy Quest	5	6	9784389	2450
5	Toy Story 2	6	5	8431650	2752
6	Magnolia	7	29	7429087	1034
7	Deuce Bigalow: Male Gigolo	8	8	6354336	2066
8	Bicentennial Man	9	7	6245638	2612
9	Snow Falling on Cedars	10	50	5117555	1150

In [3]:

```
#INGESTING PRODUCTS DATA INTO PRODUCTS

s3_client = boto3.client("s3")

BUCKET='ejcinemas'
KEY='products/products.csv'


response = s3_client.get_object(Bucket=BUCKET, Key=KEY)
products = pd.read_csv(response.get("Body"))
products.head(10)
```

Out[3]:

	Unnamed: 0	ID	Title	Publisher	Release_Date	Summary	Director
0	0	1	9.99	Regent Releasing	unknown	Have you ever wondered "What is the meaning of...	Tatia Rosenthal
1	1	2	$pent	Regent Releasing	unknown	This comic drama examines the relationships an...	Gil Cates Jr.
2	2	3	'R Xmas	Pathfinder Pictures	8-Nov-02	It's a few days before Christmas, and a Latin ...	Abel Ferrara
3	3	4	(500) Days of Summer	Fox Searchlight Pictures	17-Jul-09	After it looks as if she's left his life for g...	Marc Webb
4	4	5	1	IFC Midnight	unknown	NaN	NaN
5	5	6	...And They Lived Happily Ever After	Kino International	8-Apr-05	What makes a marriage? Georges and Natalie arg...	Yvan Attal
6	6	7	...So Goes the Nation	IFC First Take	4-Oct-06	This documentary examines America's tumultuous...	NaN
7	7	8	10 Items or Less	Click Star	1-Dec-06	A well-known actor, who hasn't accepted a role...	Brad Silberling

	Unnamed: 0	ID	Title	Publisher	Release_Date	Summary	Director
8	8	9	10 Things I Hate About You	Buena Vista Pictures	31-Mar-99	Adapted from William Shakespeare's play "The T...	Gil Junger
9	9	10	10 Years	Anchor Bay Entertainment	14-Sep-12	"10 Years" follows a group of friends on the n...	Jamie Linden

In [4]:

```
#INGESTING USERS DATA INTO USERS

s3_client = boto3.client("s3")

BUCKET='ejcinemas'
KEY='users/users.csv'


response = s3_client.get_object(Bucket=BUCKET, Key=KEY)
users= pd.read_csv(response.get("Body"))
users.head(10)
```

Out[4]:

	ID_Movie	User	Score	Review	Sentiment
0	1	DemiRonin	7	This review contains spoilers, click expand to...	12
1	1	steven	4	I don't mean to be a Debbie Downer and I am al...	2
2	3	RayJ.	9	Superb.	0
3	3	MichaelV.	9	Lillo is so hot!	0
4	3	GilbertMulroneycakesAndFriends	6	What the hell is that title all about? I assum...	-4
5	3	Ice-T	10	My movies rock!	0
6	4	Swati	7	It had its moments. I could not shake off the ...	-4
7	4	applesandorange	7	This movie is unique and not like any other lo...	5
8	4	Famousdog	10	I loooove coming into a film with absolutely n...	7
9	4	drlowdon	8	Starring Joseph Gordon-Levitt and the lovely Z...	17

In [90]:

```
#Checking the sizes's of each dataset
print(critics.shape)
print(sale.shape)
print(users.shape)
print(products.shape)
```

```
(148640, 6)
(118831, 5)
(286359, 5)
(10781, 19)
```

In [91]:

```
critics.describe()
```

Out[91]:

	Unnamed: 0	ID_Movie	Sentiment
count	148640.000000	148640.000000	148639.000000
mean	74319.500000	3548.488778	0.742679
std	42908.816343	2108.596688	2.731062
min	0.000000	1.000000	-20.000000
25%	37159.750000	1758.000000	0.000000
50%	74319.500000	3502.000000	0.000000
75%	111479.250000	5222.000000	2.000000
max	148639.000000	7763.000000	23.000000

In [92]:

```
sale.describe()
```

Out[92]:

	Rank_data	GrossW
count	118831.000000	1.188310e+05
mean	60.372386	1.676755e+06
std	36.597551	8.301432e+06
min	1.000000	1.000000e+00
25%	29.000000	4.957000e+03
50%	58.000000	3.417800e+04
75%	90.000000	2.675940e+05
max	164.000000	4.738946e+08

In [93]:

```
users.describe()
```

Out[93]:

	ID_Movie	Score	Sentiment
count	286359.000000	286359.000000	286359.000000
mean	5073.127043	6.702803	4.454066
std	2946.561907	3.219091	9.185554
min	1.000000	0.000000	-1816.000000
25%	2645.000000	5.000000	0.000000
50%	4906.000000	8.000000	3.000000
75%	7460.000000	10.000000	7.000000
max	10781.000000	10.000000	126.000000

In [94]:

```
products.describe()
```

Out[94]:

	Unnamed: 0	ID	Metascore	Meta_Pos_Count	Meta_Neut_Cour
count	10781.00000	10781.00000	10757.000000	10751.000000	10751.00000
mean	5390.00000	5391.00000	58.156270	11.669984	6.71332
std	3112.35096	3112.35096	17.406426	10.106885	5.49398
min	0.00000	1.00000	1.000000	0.000000	0.00000
25%	2695.00000	2696.00000	46.000000	4.000000	3.00000
50%	5390.00000	5391.00000	60.000000	9.000000	5.00000
75%	8085.00000	8086.00000	71.000000	17.000000	10.00000
max	10780.00000	10781.00000	100.000000	58.000000	37.00000

In [95]:

```
critics.isna().sum()
```

Out[95]:

```
Unnamed: 0    0
ID_Movie      0
Expert        0
Score         1
Review        7
Sentiment     1
dtype: int64
```

In [96]:

```
users.isna().sum()
```

Out[96]:

```
ID_Movie       0
User           5
Score          0
Review       109
Sentiment      0
dtype: int64
```

In [97]:

```
sale.isna().sum()
```

Out[97]:

```
MovieName           0
Rank_data           0
PreviousWeekRank    0
GrossW              0
Theaters            0
dtype: int64
```

In [98]:

```
products.isna().sum()
```

Out[98]:

```
Unnamed: 0            0
ID                    0
Title                 0
Publisher           288
Release_Date         24
Summary             526
Director            614
Starring           1855
Genre                29
Rating             1951
Runtime             786
Metascore            24
Meta_Pos_Count       30
Meta_Neut_Count      30
Meta_Neg_Count       30
User_Score           24
User_Pos_Count      263
User_Neut_Count     409
User_Neg_Count     1113
dtype: int64
```

In [99]:

```
#UNIQUE VALUES IN EACH VARIABLE
critics.nunique()
```

Out[99]:

```
Unnamed: 0    148640
ID_Movie        6879
Expert           187
Score            160
Review        147877
Sentiment         40
dtype: int64
```

In [100]:

```
#UNIQUE VALUES IN EACH VARIABLE
users.nunique()
```

Out[100]:

```
ID_Movie       8259
User          78910
Score            11
Review       222325
Sentiment       165
dtype: int64
```

In [101]:

```
#UNIQUE VALUES IN EACH VARIABLE
sale.nunique()
```

Out[101]:

```
MovieName           11557
Rank_data             164
PreviousWeekRank      161
GrossW              78275
Theaters             4005
dtype: int64
```

In [103]:

```
#UNIQUE VALUES IN EACH VARIABLE
products.nunique()
```

Out[103]:

```
Unnamed: 0         10781
ID                 10781
Title              10597
Publisher           1027
Release_Date        2192
Summary            10254
Director            5324
Starring            8920
Genre               1652
Rating              3555
Runtime              177
Metascore             98
Meta_Pos_Count        58
Meta_Neut_Count       36
Meta_Neg_Count        30
User_Score            93
User_Pos_Count       486
User_Neut_Count       72
User_Neg_Count       149
dtype: int64
```

CREATING TABLES WITH ATHENA

In [5]:

```
#Locating the S3 bucket

!aws s3 ls s3://ejcinemas/critics/
!aws s3 ls s3://ejcinemas/sale/
!aws s3 ls s3://ejcinemas/users/
!aws s3 ls s3://ejcinemas/products/
```

```
2022-03-24 02:33:44          0
2022-03-24 02:33:59   27983872 critics.csv
2022-03-24 02:34:48          0
2022-03-24 02:35:11    3983441 sale.csv
2022-03-24 02:35:32          0
2022-03-24 02:35:47  154997685 users.csv
2022-03-24 02:38:50          0
2022-03-24 02:39:28    9836997 products.csv
```

In [6]:

```
import boto3
import sagemaker
import pandas as pd

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name
account_id = boto3.client("sts").get_caller_identity().get("Account")

sm = boto3.Session().client(service_name="sagemaker", region_name=regio
n)
```

In [7]:

```
#SET S3 SOURCE LOCATION (PUBLIC S3 BUCKET)

s3_public_path_csv = "s3://ejcinemas"
%store s3_public_path_csv
```

```
Stored 's3_public_path_csv' (str)
```

In [8]:

```
#SET S3 DESTINATION LOCATION (PRIVATE S3 BUCKET)

s3_private_path_csv = "s3://{}/ejcinemas".format(bucket)
print(s3_private_path_csv)
```

```
s3://sagemaker-us-east-1-054298365223/ejcinemas
```

In [9]:

```
#COPY DATA FROM PUBLIC BUCKET S3 TO OUR PRIVATE S3 BUCKET
!aws s3 cp --recursive $s3_public_path_csv/ $s3_private_path_csv/ --exc
lude "*" --include "critics/critics.csv"
!aws s3 cp --recursive $s3_public_path_csv/ $s3_private_path_csv/ --exc
lude "*" --include "sale/sale.csv"
!aws s3 cp --recursive $s3_public_path_csv/ $s3_private_path_csv/ --exc
lude "*" --include "users/users.csv"
!aws s3 cp --recursive $s3_public_path_csv/ $s3_private_path_csv/ --exc
lude "*" --include "products/products.csv"
```

```
copy: s3://ejcinemas/critics/critics.csv to s3://sagemaker-u
s-east-1-054298365223/ejcinemas/critics/critics.csv
copy: s3://ejcinemas/sale/sale.csv to s3://sagemaker-us-east
-1-054298365223/ejcinemas/sale/sale.csv
copy: s3://ejcinemas/users/users.csv to s3://sagemaker-us-ea
st-1-054298365223/ejcinemas/users/users.csv
copy: s3://ejcinemas/products/products.csv to s3://sagemaker
-us-east-1-054298365223/ejcinemas/products/products.csv
```

In [10]:

```
print(s3_private_path_csv)
```

```
s3://sagemaker-us-east-1-054298365223/ejcinemas
```

In [11]:

```
!aws s3 ls $s3_private_path_csv/critics/
!aws s3 ls $s3_private_path_csv/sale/
!aws s3 ls $s3_private_path_csv/users/
!aws s3 ls $s3_private_path_csv/products/
```

```
2022-03-24 03:46:42   27983872 critics.csv
2022-03-24 03:46:44    3983441 sale.csv
2022-03-24 03:46:45  154997685 users.csv
2022-03-24 03:46:46    9836997 products.csv
```

In [12]:

```
!pip install --disable-pip-version-check -q PyAthena==2.1.0

from pyathena import connect
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypt
o.py:16: CryptographyDeprecationWarning: int_from_bytes is d
eprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.p
y:25: CryptographyDeprecationWarning: int_from_bytes is depr
ecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
WARNING: Running pip as the 'root' user can result in broken
permissions and conflicting behaviour with the system packag
e manager. It is recommended to use a virtual environment in
stead: https://pip.pypa.io/warnings/venv
```

In [13]:

```
# Set S3 staging directory -- this is a temporary directory used for At
hena queries
s3_private_path_critics = "s3://{}/ejcinemas/critics/".format(bucket)
print(s3_private_path_critics)

s3_private_path_sale = "s3://{}/ejcinemas/sale/".format(bucket)
print(s3_private_path_sale)

s3_private_path_users = "s3://{}/ejcinemas/users/".format(bucket)
print(s3_private_path_users)

s3_private_path_products = "s3://{}/ejcinemas/products/".format(bucket)
print(s3_private_path_products)
```

```
s3://sagemaker-us-east-1-054298365223/ejcinemas/critics/
s3://sagemaker-us-east-1-054298365223/ejcinemas/sale/
s3://sagemaker-us-east-1-054298365223/ejcinemas/users/
s3://sagemaker-us-east-1-054298365223/ejcinemas/products/
```

In [14]:

```
s3_staging_dir = "s3://{0}/athena/staging".format(bucket)

conn = connect(region_name=region, s3_staging_dir=s3_staging_dir)
```

In [15]:

```
database_name = "ejcinemas"

statement = "CREATE DATABASE IF NOT EXISTS {}".format(database_name)
print(statement)

pd.read_sql(statement, conn)

statement = "SHOW DATABASES"

df_show = pd.read_sql(statement, conn)
df_show.head(5)
```

```
CREATE DATABASE IF NOT EXISTS ejcinemas
```

Out[15]:

	database_name
0	default
1	dsoaws
2	ecinemas
3	ecinemas_sales
4	ejcinemas

In [16]:

```
#SETTING UP ATHENA PARAMETERS
database_name = 'ejcinemas'
critics_table = "critics"
sale_table = "sale"
users_table = "users"
products_table = "products"
```

In [17]:

```
#SQL STATEMENT FOR SALE TABLE

statement_sale = """CREATE EXTERNAL TABLE IF NOT EXISTS {}.{}(
         MovieName string,
         Rank_data int,
         PreviousWeekRank int,
         GrossW int,
         Theaters int


)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\\n'
LOCATION '{}'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(
    database_name, sale_table, s3_private_path_sale
)

pd.read_sql(statement_sale, conn)
```

Out[17]:

_

In [18]:

```
#GENERAL QUERY FOR SALE TABLE TO MAKE SURE IT'S WORKING PROPERLY

MovieName = "The Green Mile"

statement = """SELECT * FROM {}.{}
    WHERE MovieName = '{}' """.format(
    database_name, sale_table, MovieName
)

print(statement)

df = pd.read_sql(statement, conn)
df.head(5)
```

```
SELECT * FROM ejcinemas.sale
    WHERE MovieName = 'The Green Mile'
```

Out[18]:

	moviename	rank_data	previousweekrank	grossw	theaters
0	The Green Mile	2	3	12521303	2678
1	The Green Mile	5	2	10116614	2483
2	The Green Mile	5	5	6931088	2483
3	The Green Mile	5	5	5250826	2371
4	The Green Mile	6	5	4935743	2335

In [19]:

```
s3_staging_dir = "s3://{0}/athena/staging".format(bucket)

conn = connect(region_name=region, s3_staging_dir=s3_staging_dir)
```

In [55]:

```
critics.head()
```

Out[55]:

	Unnamed: 0	ID_Movie	Expert	Score	Review	Sentiment
0	0	7369	RogerEbert.com	88	Call Me Lucky will be an especially grueling r...	3.0
1	1	7369	New York Daily News	80	Angry, quixotic, tragic, heroic — Crimmins' li...	1.0
2	2	7369	Village Voice	80	Call Me Lucky is a loving but fair portrait of...	7.0
3	3	7369	TheWrap	75	There should be more Crimmins performance foot...	2.0
4	4	7369	Movie Nation	75	Call Me Lucky is another of those "the funnies...	3.0

In [58]:

```
#SQL STATEMENT FOR CRITICS TABLE

statement_critics = """CREATE EXTERNAL TABLE IF NOT EXISTS {}.{}(

         Unnamed int,
         ID_Movie int,
         Expert string,
         Score int,
         Review string,
         Sentiment float
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
LOCATION '{}'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(
 database_name, critics_table, s3_private_path_critics
)

pd.read_sql(statement_critics, conn)
```

Out[58]:

—

In [74]:

```
##SQL STATEMENT FOR USERS TABLE


statement_users = """CREATE EXTERNAL TABLE IF NOT EXISTS {}.{}(
         ID_Movie int,
         User string,
         Score int,
         Review string,
         Sentiment int
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n'
LOCATION '{}'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(
 database_name, users_table, s3_private_path_users
)

pd.read_sql(statement_users, conn)
```

Out[74]:

—

In [75]:

```
#SQL STATEMENT FOR PRODUCTS TABLE

statement_products = """CREATE EXTERNAL TABLE IF NOT EXISTS {}.{}(
     ID int,
     Title string,
     Publisher string,
     Release_Date string,
     Summary string,
     Director string,
     Starring string,
     Genre string,
     Rating string,
     Runtime string,
     Metascore int,
     Meta_Pos_Count int,
     Meta_Neut_Count int,
     Meta_Neg_Count int,
     User_Score int,
     User_Pos_Count int,
     User_Neut_Count int,
     User_Neg_Count int
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES TERMINATED BY '\\n'
LOCATION '{}'
TBLPROPERTIES ('skip.header.line.count'='1')""".format(
 database_name, products_table, s3_private_path_products
)

pd.read_sql(statement_products, conn)
```

Out[75]:

—

In [76]:

```
#CHECKING TO SEE IF ALL TABLES ARE CREATED

statement = "SHOW TABLES in {}".format(database_name)

df_show = pd.read_sql(statement, conn)
df_show.head(5)
```

Out[76]:

	tab_name
0	critics
1	products
2	sale
3	users