CIS 4130 CMWA

Professor Richard D Holowczak
Baruch College
Salman Ahmed
salman.ahmed@baruchmail.cuny.edu

Proposal

Amazon is a multinational technology company focusing on e-commerce, cloud computing, online advertising, digital streaming, and artificial intelligence. Jeff Bezos founded Amazon from his garage in Bellevue, Washington on July 5, 1994. The company started as an online marketplace for books, and over the year it has expanded into a multitude of product categories site. I will be working with Amazon Customer Review Dataset. Amazon has 310 million customers worldwide and ships approximately 1.6 million packages daily. The customers can write reviews about the product they received to share with other people shopping on amazon. I will use the amazon customer review dataset for this project to find what makes a great product and best seller for the different shopping categories.

In this dataset, we have attributes like a marketplace, product_title, product_category, star_rating, helpful_votes, and more. Each of these attributes plays a role in helping identify a great product for each shopping category. As the reviews of 1 product in different multiple categories may differ for the different marketplaces, which leads to different star_rating. Using review attributes like review_headline and review_body, we can predict the rating ourselves using word lists.

The data set can be found here: https://s3.amazonaws.com/amazon-reviews-pds/readme.html

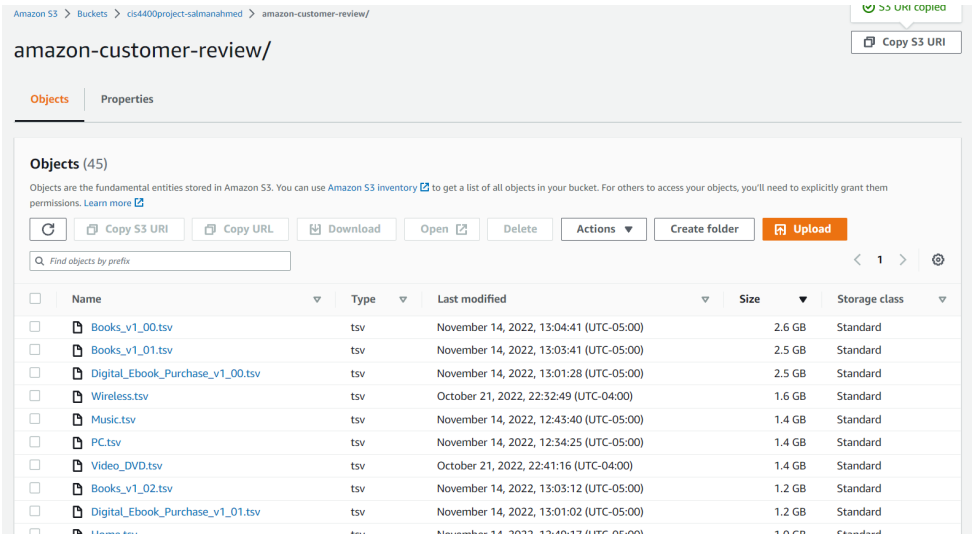
Data Acquisition

I was able to find the data through the recommendation of my professor. As I was debating whether to work on the project that works solo with headphones reviews, Professor Richard D Holowczak wanted me to check out the dataset of Amazon reviews. This dataset is provided by amazon itself.

I created an AWS bucket S3 on Command Line Interface (CLI), which can be accessed through the EC2 instance in the AWS web service. To create the bucket first, I used this command line: *$ aws s3api create-bucket --bucket cis4400project-salmanahmed --region us-east-2 --create-bucket-configuration \ LocationConstraint=us-east-2.* After the S3 bucket is created, I had to migrate the data from [Amazon Customer reviews] into the bucket. Since the file is already hosted on S3, I can copy it to

aws s3 cp s3://amazon-reviews-pds/tsv/amazon_reviews_us_Wireless_v1_00.tsv.gz s3://cis4400project-salmanahmed

We have so many different types of review reviews in this from watches to Office Products and more. s3://cis4400project-salmanahmed/amazon-customer-review/



Milestone 3

We have the data in the S3 bucket, before running any models or working on the data, we need to clean the data. I wanted to see if there are any columns that have NULL values.

```
>>> sdf.select([count(when(col(c).isNull(), c)).alias(c) for c in ["review_date"]] ).show()
+-----------+
|review_date|
+-----------+
|          0|
+-----------+
```

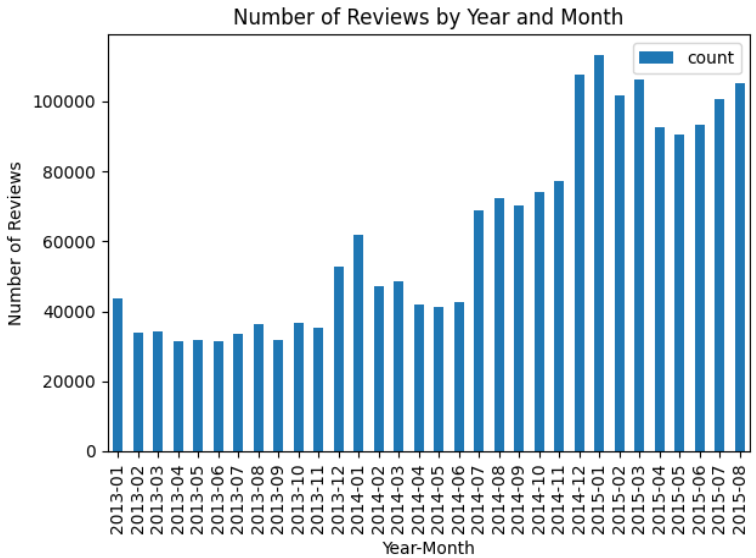
I wanted to get a count of the records I have to work with

```
>>> sdf.count()
2391152
```

This is where I looked at the Look at some of the statistics for some specific columns. Stats like count, min, max, and mean to get an idea of what I am working with

```
>>> sdf.select("star_rating","helpful_votes","total_votes").summary("count", "min", "max", "mean").show()
+-------+------------------+------------------+------------------+
|summary|       star_rating|      helpful_votes|       total_votes|
+-------+------------------+------------------+------------------+
|  count|           2391152|           2391152|           2391152|
|    min|                 1|                 0|               0.0|
|    max|                 5|              6405|            6520.0|
|   mean|4.159210288597295|1.8417507544480651|2.2050630825643873|
+-------+------------------+------------------+------------------+
```

Using pandas I plotted a graph to see the reviews by Year and month. I grouped them and got to get a count by year and month. Converted the grouped data into a pandas data frame.



Milestone 4 and 5

I wanted to create an AWS EMR cluster with PySpark code to read and process this data

```
from pyspark.sql.functions import *
from pyspark.ml.feature import StringIndexer, OneHotEncoder, VectorAssembler
from pyspark.ml import Pipeline
from pyspark.ml.classification import LogisticRegression
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.sql.types import DoubleType
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

I used 3 files for my file_list from the S3 bucket. I will be exploring these files and training them.

```
bucket = 's3a://amazon-reviews-pds/tsv/'
file_list = ['s3a://amazon-reviews-pds/tsv/amazon_reviews_us_Home_Entertainment_v1_00.tsv.gz',
             's3a://amazon-reviews-pds/tsv/amazon_reviews_us_Home_v1_00.tsv.gz',
             's3a://amazon-reviews-pds/tsv/amazon_reviews_us_Home_Improvement_v1_00.tsv.gz']
```

I split the data into training and test sets

```
trainingData, testData = sdf.randomSplit([0.7, 0.3], seed=3456)
```

I created an indexer for the three strings based on columns: "product_category", "vine", and “verified_purchase". With that, I created a LogisticRegression Estimator and trained the models. I used AUC to find out the e-value of AUC that characterizes the model performance. Higher the AUC value, the higher performance of the model. The perfect classifier will have a high value of true positive rate and a low value of false positive.

```
Number of models to be tested:  18
>>> evaluator = BinaryClassificationEvaluator(metricName="areaUnderROC")
>>> cv = CrossValidator(estimator=reviews_pipe,
... estimatorParamMaps=grid,
... evaluator=evaluator,
... numFolds=3,
... seed=789
... )
```

```
>>> print('AUC:', auc)
AUC: 0.6122850201888945
>>>
```

```
>>> predictions.groupby('label').pivot('prediction').count().fillna(0).show()
+-----+---+------+
|label|0.0|   1.0|
+-----+---+------+
|  1.0| 23|558466|
|  0.0| 12|159763|
+-----+---+------+
```

Classification accuracy is the total number of correct predictions divided by the total number of predictions made for a dataset.

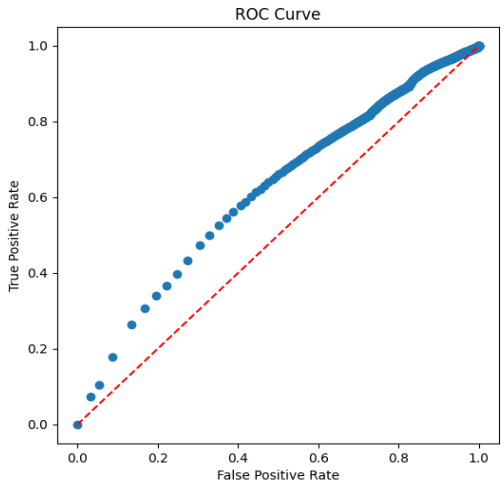
Precision = TruePositives / (true positives + false positives)

```
>>> print( calculate_precision_recall(cm) )
(0.22246137910294822, 0.22244019665037196, 0.9999248943827257, 0.36392317119284195)
>>>
```

In this case, although the model predicted far higher examples belonging to the minority class, the ratio of correct positive examples is much lower.

I wanted to create a ROC curve (receiver operating characteristic curve) showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.

```
>>> parammap = cv.bestModel.stages[3].extractParamMap()
>>> for p, v in parammap.items():
...     print(p, v)
...
LogisticRegression_1772057b4847__aggregationDepth 2
LogisticRegression_1772057b4847__elasticNetParam 0.0
LogisticRegression_1772057b4847__family auto
LogisticRegression_1772057b4847__featuresCol features
LogisticRegression_1772057b4847__fitIntercept True
LogisticRegression_1772057b4847__labelCol label
LogisticRegression_1772057b4847__maxBlockSizeInMB 0.0
LogisticRegression_1772057b4847__maxIter 10
LogisticRegression_1772057b4847__predictionCol prediction
LogisticRegression_1772057b4847__probabilityCol probability
LogisticRegression_1772057b4847__rawPredictionCol rawPrediction
LogisticRegression_1772057b4847__regParam 1.0
LogisticRegression_1772057b4847__standardization True
LogisticRegression_1772057b4847__threshold 0.5
LogisticRegression_1772057b4847__tol 1e-06
```



These are the coefficients of each of the variables, I looped them through the feature to extract and store them on the var_index dictionary. In the matrix of the relationships between each pair of variables in the dataset, the result is a symmetric matrix called a correlation matrix. 1.0 being perfectly correlated with itself and 0.0 being no correlation.

```
0 product_categoryVector_Home 0.0041058831357235455
1 product_categoryVector_Home Improvement 0.011936775906431198
2 product_categoryVector_Home Entertainment -0.04856530131342955
3 product_categoryVector___unknown 0.0
4 vineVector_N -0.06661166751853127
5 vineVector_Y 0.06661166751577872
6 vineVector___unknown 0.0
7 verified_purchaseVector_Y 0.07663812278440076
8 verified_purchaseVector_N -0.0766381227843193
9 verified_purchaseVector___unknown 0.0
10 total_votes -0.0007247042096972O4
11 review_body_wordcount -0.0003310000908902S5
```

Milestone 6

Working with AWS was a new thing and was having a hard time understanding the topics and reasoning behind using cloud computing. Over the time working on the project, a lot of the concepts have become clearer. When we visualize the data we get a better understanding of the issues and the bigger picture. I wish I spend more time visualizing the data. Working with Pyspark built using Python, seems a little complicated.

Milestone 7

https://github.com/SalmanAhmedDevelopment/CIS4130