CS4830 - Big Data Lab

Final Project

Classification on the Yelp dataset using NLP

Akash Anand - ED18B002

Nihal S Manvi - AE18B031

Shashank H S - BE18B006

Indian Institute of Technology Madras

Jan - May 2022

Contents

List of Figures

1 Introduction

In this project, we are provided with a real-world dataset upon which we perform the required analysis using tools and services learnt during the course on the Google Cloud Platform. The dataset that we are provided with is a subset of data from the Yelp website which contains many user-submitted reviews of various businesses.

2 Problem Statement

The dataset provided to us is over 3 GB large. Our aim is to classify a review into a given star-rating category by performing sentiment analysis using NLP on the review text.

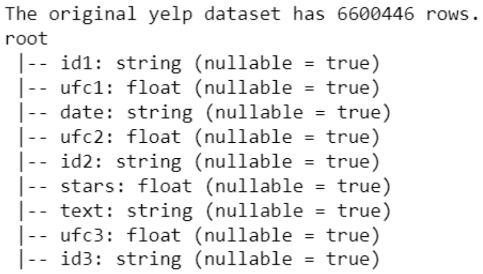
Our approach was to pre-process the provided data and train several machine learning models on it to come up with the predictions. We use Kafka streaming model for testing the model and making predictions in the subscriber on data coming in from the publisher. The required steps that have been performed in this project are as follows:

1. Exploratory Data Analysis
2. Preprocessing and Feature Engineering
3. Training of different models
4. Accuracy/F1-score comparisons
5. Kafka streaming

3 Exploratory Data Analysis and Data Preprocessing

Default image version for cluster is Debian 10, Hadoop 3.2, Spark 3.1. Due to Anaconda's unavailability with these specifications, we changed the cluster image version to Ubuntu 18.04 LTS, Hadoop 2.10, Spark 2.4.

A schema of the dataset is shown in the figure below.



```
The original yelp dataset has 6600446 rows.
root
 |-- id1: string (nullable = true)
 |-- ufc1: float (nullable = true)
 |-- date: string (nullable = true)
 |-- ufc2: float (nullable = true)
 |-- id2: string (nullable = true)
 |-- stars: float (nullable = true)
 |-- text: string (nullable = true)
 |-- ufc3: float (nullable = true)
 |-- id3: string (nullable = true)
```

Figure 1: Schema of Yelp Data

Some more details about the various features in the provided csv follow. Three input labels are given as id1, id2, id3 which represent three unique identifiers:

1. User ID: Unique ID provided by Yelp to a particular user

2. Business ID: Unique ID provided by Yelp to a particular business
3. Review ID: Unique ID provided by Yelp to a particular review

With the provided csv file, it is unclear which id column corresponds to which exact id.

Three more input labels ufc1, ufc2, ufc3 are provided which are numerical ratings for:

1. Useful - The number of people who found the review useful
2. Funny - The number of people who found the review funny
3. Cool - The number of people who found the review cool

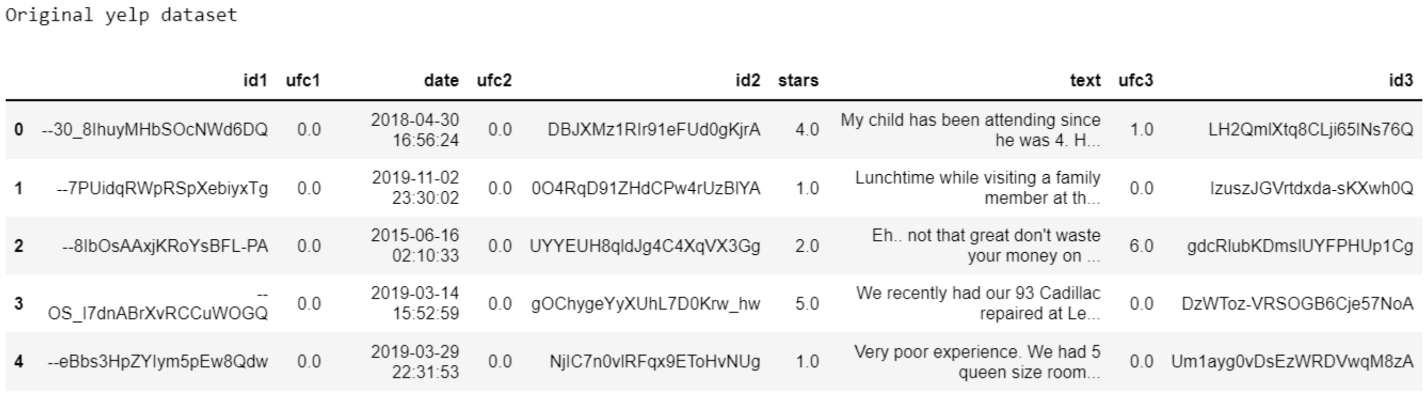
Again, it is unclear with the provided csv file as to which column corresponds to which type of rating. The above uncertainties in column name does not affect training in any capacity.

Additional input labels which are clear as per the given csv file are:

1. Date - Date on which the review was uploaded
2. Text - Text feedback provided by the user as part of the review

The output label that has been provided to us is the star rating: the rating of a particular business given by an user.

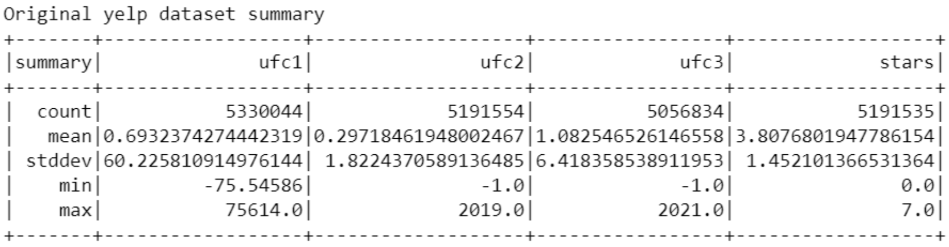
A summary of the dataset provided to us is:



Original yelp dataset

	id1	ufc1	date	ufc2	id2	stars	text	ufc3	id3
0	--30_8IhuyMHbSOcNWd6DQ	0.0	2018-04-30 16:56:24	0.0	DBJXMz1RIr91eFUd0gKjrA	4.0	My child has been attending since he was 4. H...	1.0	LH2QmIXtq8CLji65INs76Q
1	--7PUidqRWpRSpXebiyxTg	0.0	2019-11-02 23:30:02	0.0	0O4RqD91ZHdCPw4rUzBlYA	1.0	Lunchtime while visiting a family member at th...	0.0	IzuszJGVrtdxda-sKXwh0Q
2	--8IbOsAAxjKRoYsBFL-PA	0.0	2015-06-16 02:10:33	0.0	UYYEUH8qldJg4C4XqVX3Gg	2.0	Eh.. not that great don't waste your money on ...	6.0	gdcRlubKDmslUYFPHUp1Cg
3	--OS_I7dnABrXvRCCuWOGQ	0.0	2019-03-14 15:52:59	0.0	gOChygeYyXUhL7D0Krw_hw	5.0	We recently had our 93 Cadillac repaired at Le...	0.0	DzWToz-VRSOGB6Cje57NoA
4	--eBbs3HpZYIym5pEw8Qdw	0.0	2019-03-29 22:31:53	0.0	NjIC7n0vlRFqx9EToHvNUg	1.0	Very poor experience. We had 5 queen size room...	0.0	Um1ayg0vDsEzWRDVwqM8zA

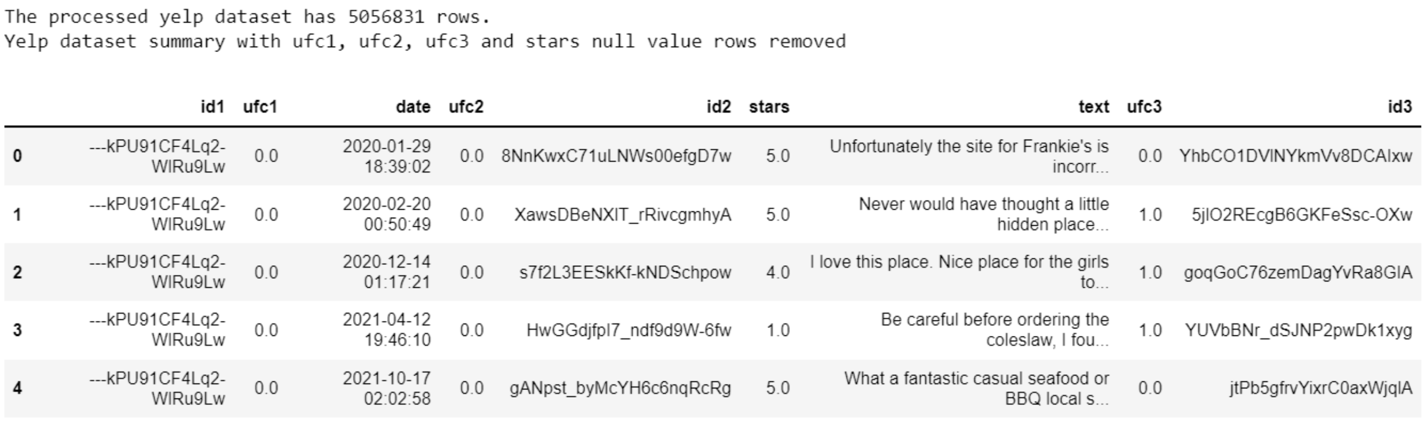
Figure 2: Yelp Data Before Processing



```
Original yelp dataset summary
+-------+------------------+-------------------+-----------------+------------------+
|summary|              ufc1|               ufc2|             ufc3|             stars|
+-------+------------------+-------------------+-----------------+------------------+
|  count|           5330044|            5191554|          5056834|           5191535|
|   mean|0.6932374274442319|0.29718461948002467|1.082546526146558|3.8076801947786154|
| stddev|60.225810914976144| 1.8224370589136485|6.418358538911953| 1.452101366531364|
|    min|         -75.54586|               -1.0|             -1.0|               0.0|
|    max|           75614.0|             2019.0|           2021.0|               7.0|
+-------+------------------+-------------------+-----------------+------------------+
```

Figure 3: Summary of Yelp Data Before Processing

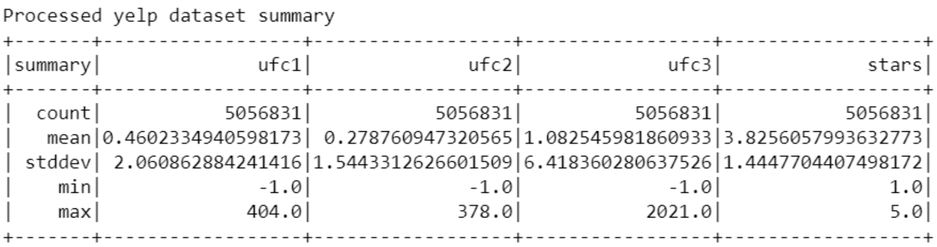
The data in the above figure came with many rows having many of the features having NaN values. As a result, such rows were naturally filtered from the dataset. A summary of the obtained data after processing is shown below:



```
The processed yelp dataset has 5056831 rows.
Yelp dataset summary with ufc1, ufc2, ufc3 and stars null value rows removed
```

	id1	ufc1	date	ufc2	id2	stars	text	ufc3	id3
0	---kPU91CF4Lq2-WIRu9Lw	0.0	2020-01-29 18:39:02	0.0	8NnKwxC71uLNWs00efgD7w	5.0	Unfortunately the site for Frankie's is incorr...	0.0	YhbCO1DVlNYkmVv8DCAIxw
1	---kPU91CF4Lq2-WIRu9Lw	0.0	2020-02-20 00:50:49	0.0	XawsDBeNXIT_rRivcgmhyA	5.0	Never would have thought a little hidden place...	1.0	5jlO2REcgB6GKFeSsc-OXw
2	---kPU91CF4Lq2-WIRu9Lw	0.0	2020-12-14 01:17:21	0.0	s7f2L3EESkKf-kNDSchpow	4.0	I love this place. Nice place for the girls to...	1.0	goqGoC76zemDagYvRa8GIA
3	---kPU91CF4Lq2-WIRu9Lw	0.0	2021-04-12 19:46:10	0.0	HwGGdjfpl7_ndf9d9W-6fw	1.0	Be careful before ordering the coleslaw, I fou...	1.0	YUVbBNr_dSJNP2pwDk1xyg
4	---kPU91CF4Lq2-WIRu9Lw	0.0	2021-10-17 02:02:58	0.0	gANpst_byMcYH6c6nqRcRg	5.0	What a fantastic casual seafood or BBQ local s...	0.0	jtPb5gfrvYixrC0axWjqlA

Figure 4: Yelp Data Post Processing



```
Processed yelp dataset summary
+-------+------------------+------------------+-----------------+------------------+
|summary|              ufc1|              ufc2|             ufc3|             stars|
+-------+------------------+------------------+-----------------+------------------+
|  count|           5056831|           5056831|          5056831|           5056831|
|   mean|0.4602334940598173| 0.278760947320565|1.082545981860933|3.8256057993632773|
| stddev| 2.060862884241416|1.5443312626601509|6.418360280637526|1.4447704407498172|
|    min|              -1.0|              -1.0|             -1.0|               1.0|
|    max|             404.0|             378.0|           2021.0|               5.0|
+-------+------------------+------------------+-----------------+------------------+
```

Figure 5: Summary of Yelp Data Post Processing

Some overarching insights we gain from the above data follow:

1. Star values range from 0 to 7 in integer increments before processing, and 0 to 5 after processing.
2. There is no particular limited range as such for useful, funny, cool variables because the value completely depends on other users on the website and their response to the reviews

3.1 Graphical Representations

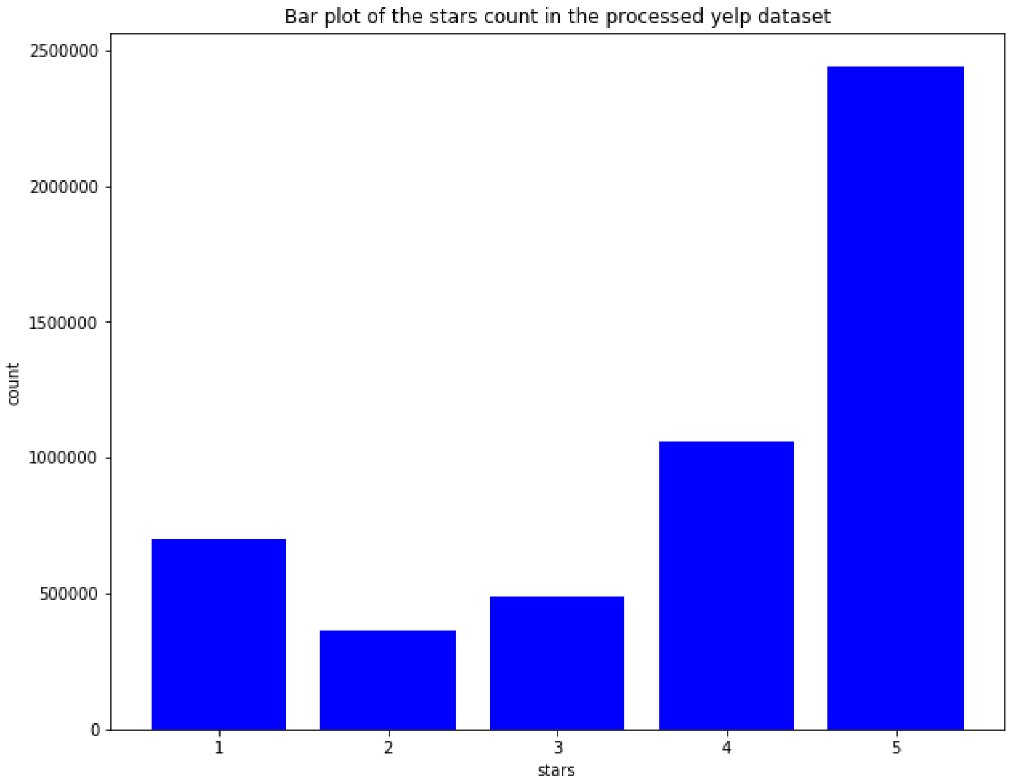


Figure 6: Bar Plot Distribution of Stars Data

From the above plot, we can see that majority of the reviews have 5-star ratings while very few have 2-star ratings.

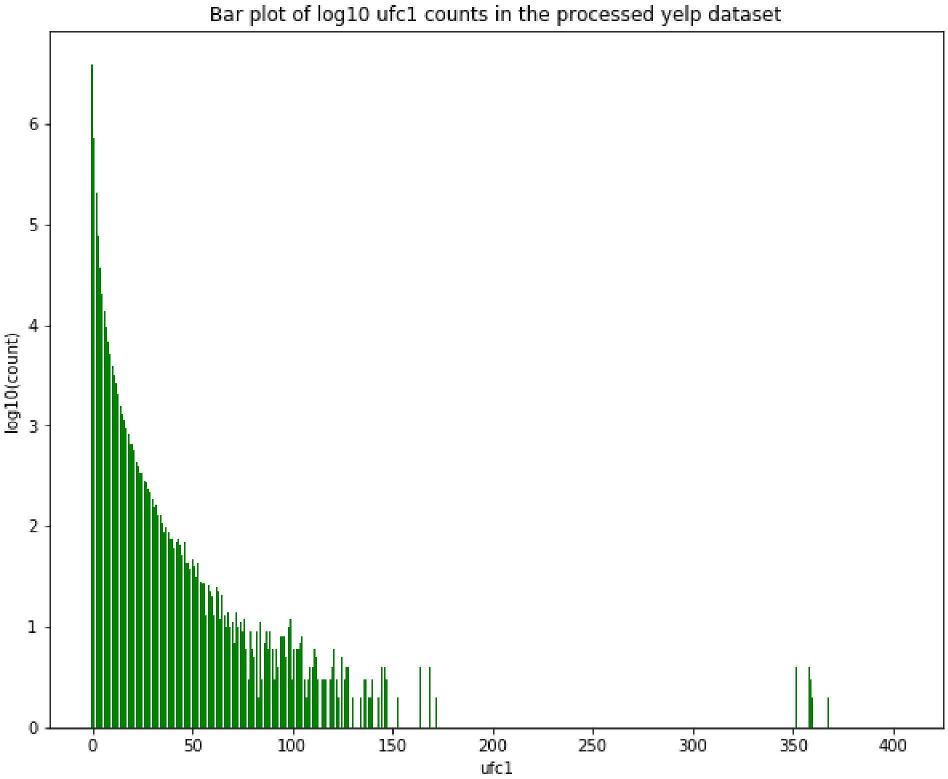


Figure 7: Bar Plot Distribution of ufc1 Data

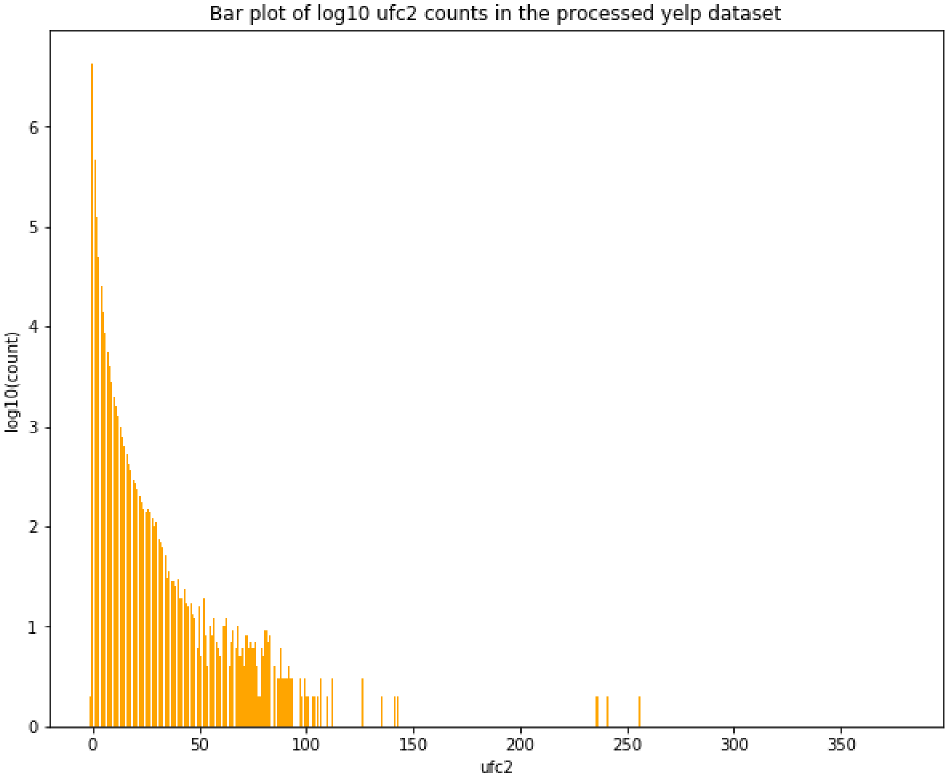


Figure 8: Bar Plot Distribution of ufc2 Data

The plots of ufc1 and ufc2 show that the data is highly skewed towards the lower end of the X axis.

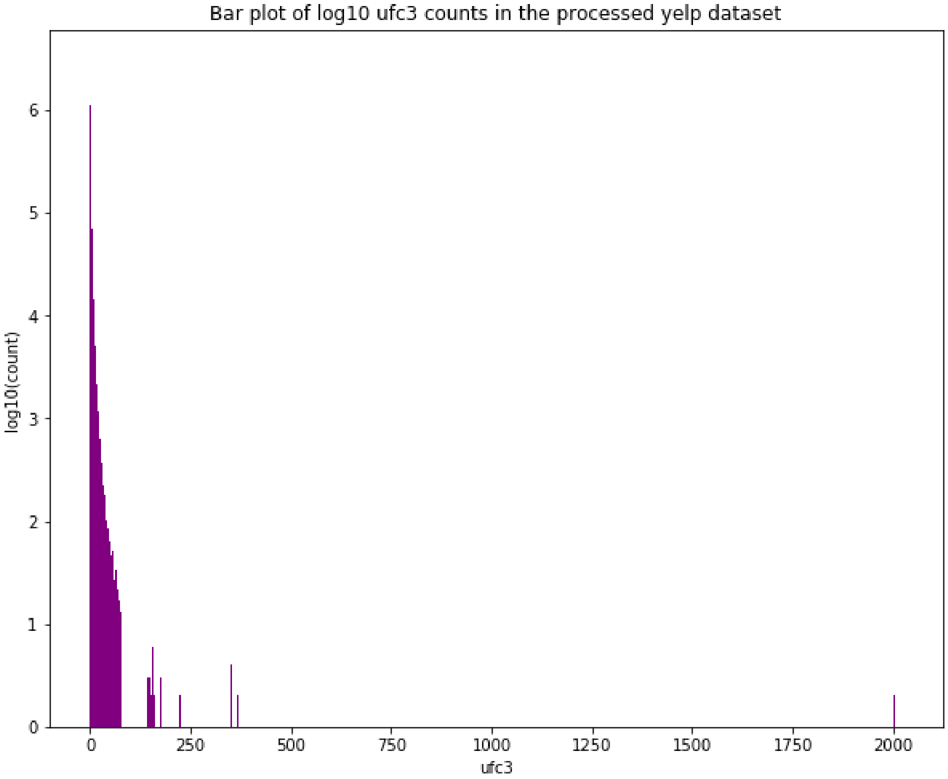
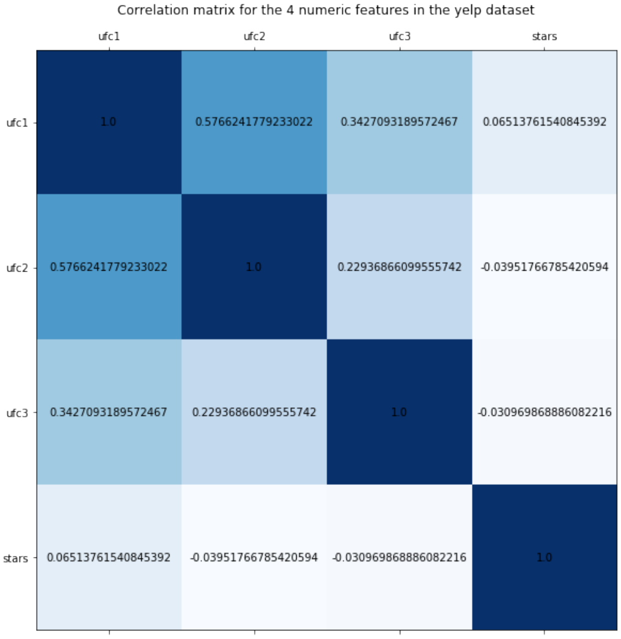


Figure 9: Bar Plot Distribution of ufc3 Data



	ufc1	ufc2	ufc3	stars
ufc1	1.0	0.5766241779233022	0.3427093189572467	0.06513761540845392
ufc2	0.5766241779233022	1.0	0.22936866099555742	-0.03951766785420594
ufc3	0.3427093189572467	0.22936866099555742	1.0	-0.030969868886082216
stars	0.06513761540845392	-0.03951766785420594	-0.030969868886082216	1.0

Figure 10: Correlation Matrix of the Features

From this correlation matrix, we can see that the given numerical input variables ufc1, ufc2, ufc3 have very low correlation with ‘stars’ variables. Thus, from this we can sense that there will be high dependency on the ‘text’ data variable for classification modelling

4 Pre-processing and Feature Engineering

The following series of operations were performed on the ‘text’ column to obtain a tf-idf vector representation of the field.

1. Document assembler - Transforms the given data into the required format for the Annotator in NLP Spark to work on
2. Tokenizer - It breaks the raw text into small chunks. Tokenization breaks the raw text into words called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of words. For example, the text “It is raining” can be tokenized into ‘It’, ‘is’, ‘raining’
3. Normalizer - Removes all dirty characters from text following a regex pattern and transforms words based on a provided dictionary. It brings all text to lowercase and removes punctuation.
4. Stemmer - Works by cutting off the end of the word, taking into account a list of common suffixes that can be found in an inflected word. This indiscriminate cutting can be successful in some occasions, but not always.
5. Finisher - This is used to close the annotation and transformation done for the NLP spark module’s usage
6. Stopword Remover - This removes all the stopwords like ‘is’, ‘a’, ‘an’, ‘not’, ‘are’, etc in the given text. This cleaner dataset helps in feature engineering

After the pre-processing the text as described above, feature engineering was done on the text using TF-IDF and Word2Vec. Since Word2Vec took too long, TF-IDF was chosen for feature engineering.

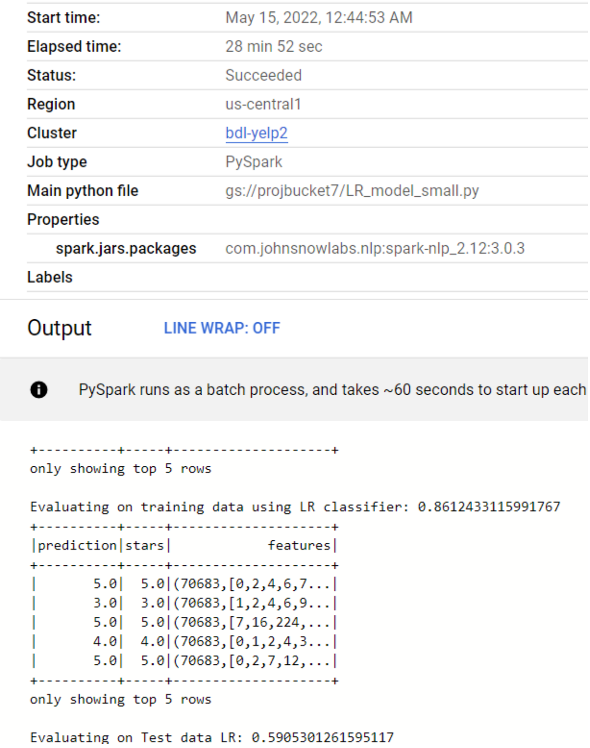
5 Training Models

After the pre-processing and feature engineering, we use the modified dataset to train logistic regression, Naive Bayes, decision tree models. Since the dataset is huge and the TF-IDF vectors obtained after preprocessing are large, choosing a complex model (such as SVM or Random Forest) would potentially result in very high training time. Therefore, we used simple models such as Naive Bayes Classifier, Logistic Regression and Decision Tree Classifier. The training is initially done on a small part of the dataset and it is observed as to which model is performing best on the subset of the data. Later, the model giving the highest accuracy is selected for training on the complete dataset.

For training, we used a Dataproc Cluster with one master node and two worker nodes (n1-standard-2 worker and n1-standard-4 master).

The results of training are given below:

5.1 Logistic Regression



Start time:	May 15, 2022, 12:44:53 AM
Elapsed time:	28 min 52 sec
Status:	Succeeded
Region	us-central1
Cluster	bdl-yelp2
Job type	PySpark
Main python file	gs://projbucket7/LR_model_small.py
Properties	
spark.jars.packages	com.johnsnowlabs.nlp:spark-nlp_2.12:3.0.3
Labels	

PySpark runs as a batch process, and takes ~60 seconds to start up each

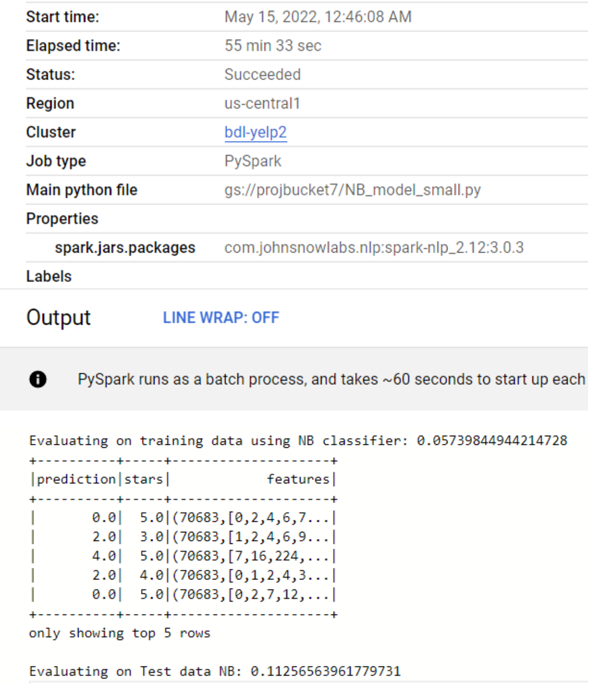
```
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on training data using LR classifier: 0.8612433115991767
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       5.0|  5.0|(70683,[0,2,4,6,7...|
|       3.0|  3.0|(70683,[1,2,4,6,9...|
|       5.0|  5.0|(70683,[7,16,224,...|
|       4.0|  4.0|(70683,[0,1,2,4,3...|
|       5.0|  5.0|(70683,[0,2,7,12,...|
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Test data LR: 0.590530126159517
```

Figure 11: Results of Logistic Regression

5.2 Naive Bayes



Start time:	May 15, 2022, 12:46:08 AM
Elapsed time:	55 min 33 sec
Status:	Succeeded
Region	us-central1
Cluster	bdl-yelp2
Job type	PySpark
Main python file	gs://projbucket7/NB_model_small.py
Properties	
spark.jars.packages	com.johnsnowlabs.nlp:spark-nlp_2.12:3.0.3
Labels	

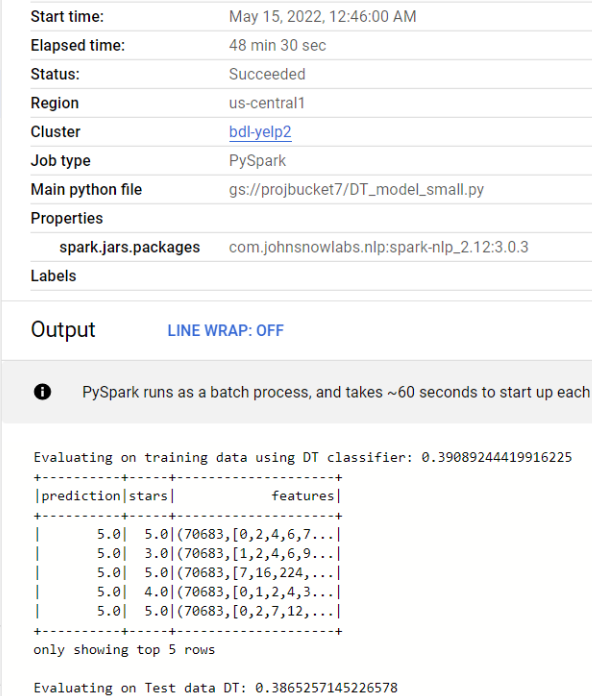
PySpark runs as a batch process, and takes ~60 seconds to start up each

```
Evaluating on training data using NB classifier: 0.05739844944214728
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       0.0|  5.0|(70683,[0,2,4,6,7...|
|       2.0|  3.0|(70683,[1,2,4,6,9...|
|       4.0|  5.0|(70683,[7,16,224,...|
|       2.0|  4.0|(70683,[0,1,2,4,3...|
|       0.0|  5.0|(70683,[0,2,7,12,...|
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Test data NB: 0.11256563961779731
```

Figure 12: Results of Naive Bayes

5.3 Decision Tree



Start time:	May 15, 2022, 12:46:00 AM
Elapsed time:	48 min 30 sec
Status:	Succeeded
Region	us-central1
Cluster	bdl-yelp2
Job type	PySpark
Main python file	gs://projbucket7/DT_model_small.py
Properties	
spark.jars.packages	com.johnsnowlabs.nlp:spark-nlp_2.12:3.0.3
Labels	

PySpark runs as a batch process, and takes ~60 seconds to start up each

```
Evaluating on training data using DT classifier: 0.39089244419916225
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       5.0|  5.0|(70683,[0,2,4,6,7...|
|       5.0|  3.0|(70683,[1,2,4,6,9...|
|       5.0|  5.0|(70683,[7,16,224,...|
|       5.0|  4.0|(70683,[0,1,2,4,3...|
|       5.0|  5.0|(70683,[0,2,7,12,...|
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Test data DT: 0.3865257145226578
```

Figure 13: Results of Decision Tree

5.4 Discussion of Results

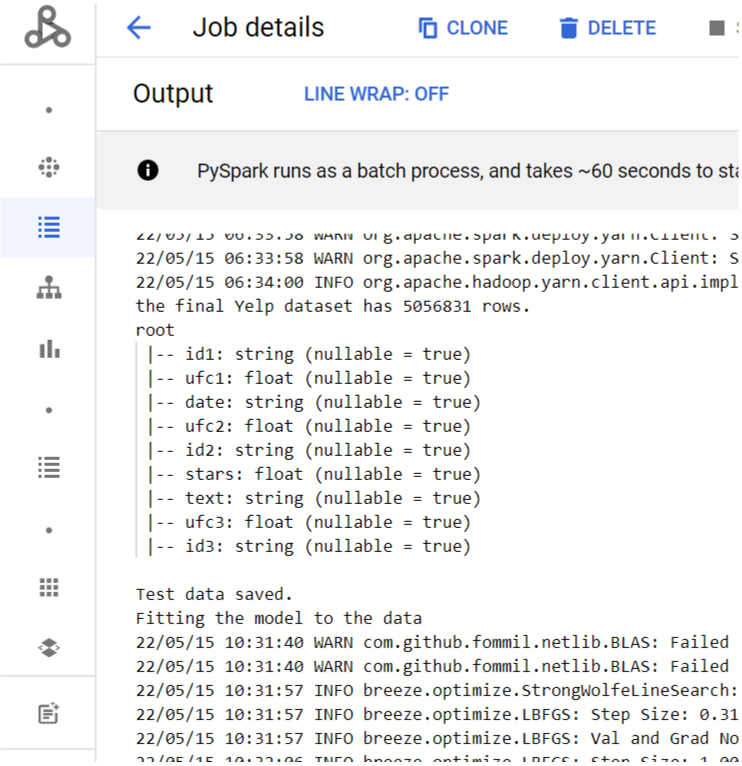
A summary of the results from the above models is shown below:

	Accuracy	
Model	Training Set	Test Set
Logistic Regression	0.861	0.590
Naive Bayes	0.057	0.112
Decision Tree	0.391	0.386

From the above results, it is evident the Logistic Regression model performs best on the train and test dataset. We use this model to proceed training the final dataset.

5.5 Final Model

Proof of training the final model is shown below:

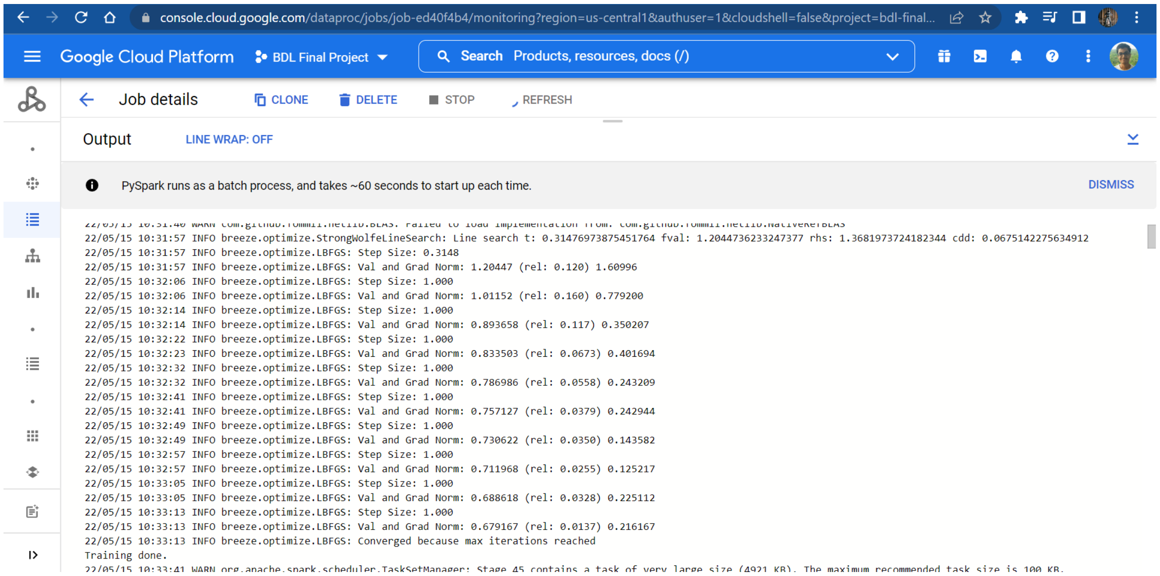


PySpark runs as a batch process, and takes ~60 seconds to sta

```
22/05/15 06:33:58 WARN org.apache.spark.deploy.yarn.Client: S
22/05/15 06:33:58 WARN org.apache.spark.deploy.yarn.Client: S
22/05/15 06:34:00 INFO org.apache.hadoop.yarn.client.api.impl
the final Yelp dataset has 5056831 rows.
root
 |-- id1: string (nullable = true)
 |-- ufc1: float (nullable = true)
 |-- date: string (nullable = true)
 |-- ufc2: float (nullable = true)
 |-- id2: string (nullable = true)
 |-- stars: float (nullable = true)
 |-- text: string (nullable = true)
 |-- ufc3: float (nullable = true)
 |-- id3: string (nullable = true)

Test data saved.
Fitting the model to the data
22/05/15 10:31:40 WARN com.github.fommil.netlib.BLAS: Failed
22/05/15 10:31:40 WARN com.github.fommil.netlib.BLAS: Failed
22/05/15 10:31:57 INFO breeze.optimize.StrongWolfeLineSearch:
22/05/15 10:31:57 INFO breeze.optimize.LBFGS: Step Size: 0.31
22/05/15 10:31:57 INFO breeze.optimize.LBFGS: Val and Grad No
22/05/15 10:32:06 INFO breeze.optimize.LBFGS: Step Size: 1.00
```

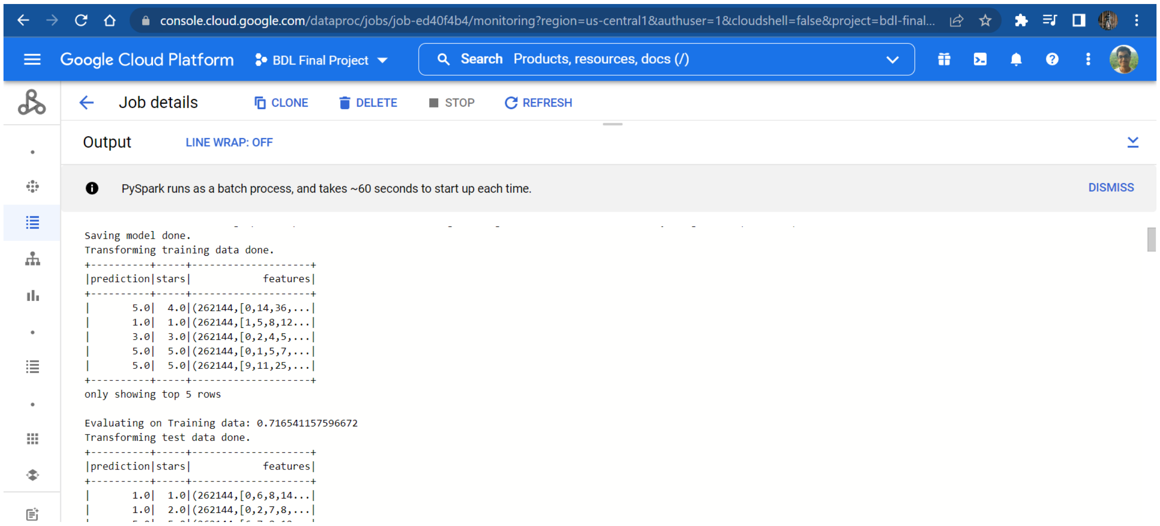
Figure 14: Training Proof 1


console.cloud.google.com/dataproc/jobs/job-ed40f4b4/monitoring?region=us-central1&authuser=1&cloudshell=false&project=bdl-final...
Google Cloud Platform
BDL Final Project
Search Products, resources, docs (/)
Job details
CLONE
DELETE
STOP
REFRESH
Output
LINE WRAP: OFF
DISMISS

PySpark runs as a batch process, and takes ~60 seconds to start up each time.

```
22/05/15 10:31:40 WARN com.github.fommil.netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
22/05/15 10:31:57 INFO breeze.optimize.StrongWolfeLineSearch: Line search t: 0.31476973875451764 fval: 1.2044736233247377 rhs: 1.3681973724182344 cdd: 0.0675142275634912
22/05/15 10:31:57 INFO breeze.optimize.LBFGS: Step Size: 0.3148
22/05/15 10:31:57 INFO breeze.optimize.LBFGS: Val and Grad Norm: 1.20447 (rel: 0.120) 1.60996
22/05/15 10:32:06 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:06 INFO breeze.optimize.LBFGS: Val and Grad Norm: 1.01152 (rel: 0.160) 0.779200
22/05/15 10:32:14 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:14 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.893658 (rel: 0.117) 0.350207
22/05/15 10:32:22 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:23 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.833503 (rel: 0.0673) 0.401694
22/05/15 10:32:32 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:32 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.786986 (rel: 0.0558) 0.243209
22/05/15 10:32:41 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:41 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.757127 (rel: 0.0379) 0.242944
22/05/15 10:32:49 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:49 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.730622 (rel: 0.0350) 0.143582
22/05/15 10:32:57 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:32:57 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.711968 (rel: 0.0255) 0.125217
22/05/15 10:33:05 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:33:05 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.688618 (rel: 0.0328) 0.225112
22/05/15 10:33:13 INFO breeze.optimize.LBFGS: Step Size: 1.000
22/05/15 10:33:13 INFO breeze.optimize.LBFGS: Val and Grad Norm: 0.679167 (rel: 0.0137) 0.216167
22/05/15 10:33:13 INFO breeze.optimize.LBFGS: Converged because max iterations reached
Training done.
22/05/15 10:33:41 WARN org.apache.spark.scheduler.TaskSetManager: Stage 45 contains a task of very large size (4921 KB). The maximum recommended task size is 100 KB.
```

Figure 15: Training Proof 2

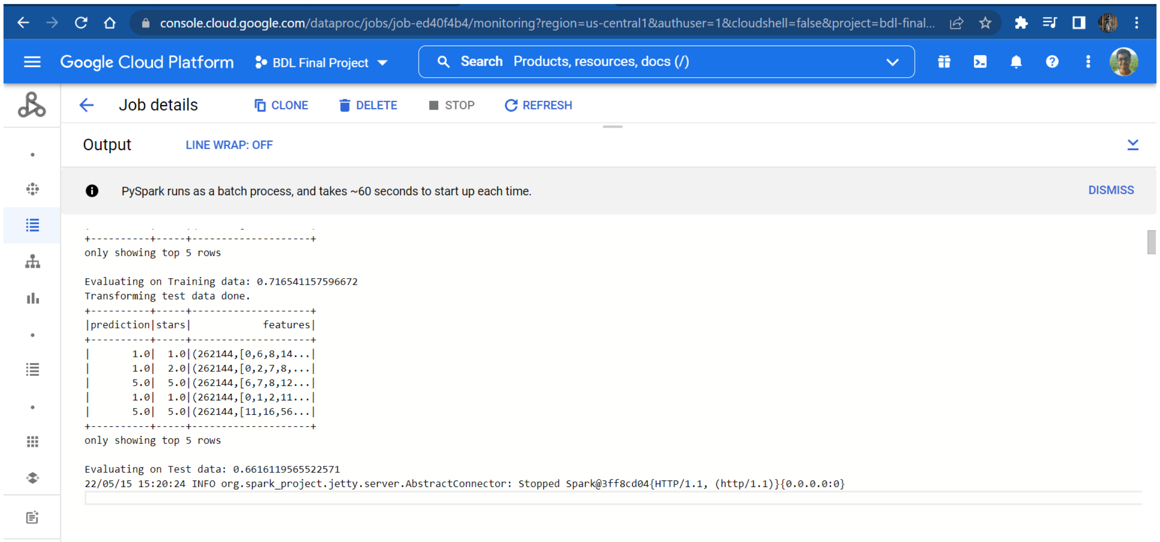

console.cloud.google.com/dataproc/jobs/job-ed40f4b4/monitoring?region=us-central1&authuser=1&cloudshell=false&project=bdl-final...
Google Cloud Platform
BDL Final Project
Search Products, resources, docs (/)
Job details
CLONE
DELETE
STOP
REFRESH
Output
LINE WRAP: OFF
DISMISS

PySpark runs as a batch process, and takes ~60 seconds to start up each time.

```
Saving model done.
Transforming training data done.
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       5.0|  4.0|(262144,[0,14,36,...|
|       1.0|  1.0|(262144,[1,5,8,12...|
|       3.0|  3.0|(262144,[0,2,4,5,...|
|       5.0|  5.0|(262144,[0,1,5,7,...|
|       5.0|  5.0|(262144,[9,11,25,...|
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Training data: 0.716541157596672
Transforming test data done.
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       1.0|  1.0|(262144,[0,6,8,14...|
|       1.0|  2.0|(262144,[0,2,7,8,...|
```

Figure 16: Training Set Result


console.cloud.google.com/dataproc/jobs/job-ed40f4b4/monitoring?region=us-central1&authuser=1&cloudshell=false&project=bdl-final...
Google Cloud Platform
BDL Final Project
Search Products, resources, docs (/)
Job details
CLONE
DELETE
STOP
REFRESH
Output
LINE WRAP: OFF
DISMISS

PySpark runs as a batch process, and takes ~60 seconds to start up each time.

```
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Training data: 0.716541157596672
Transforming test data done.
+----------+-----+--------------------+
|prediction|stars|            features|
+----------+-----+--------------------+
|       1.0|  1.0|(262144,[0,6,8,14...|
|       1.0|  2.0|(262144,[0,2,7,8,...|
|       5.0|  5.0|(262144,[6,7,8,12...|
|       1.0|  1.0|(262144,[0,1,2,11...|
|       5.0|  5.0|(262144,[11,16,56...|
+----------+-----+--------------------+
only showing top 5 rows

Evaluating on Test data: 0.6616119565522571
22/05/15 15:20:24 INFO org.spark_project.jetty.server.AbstractConnector: Stopped Spark@3ff8cd04{HTTP/1.1, (http/1.1)}{0.0.0.0:0}
```

Figure 17: Test Set Result

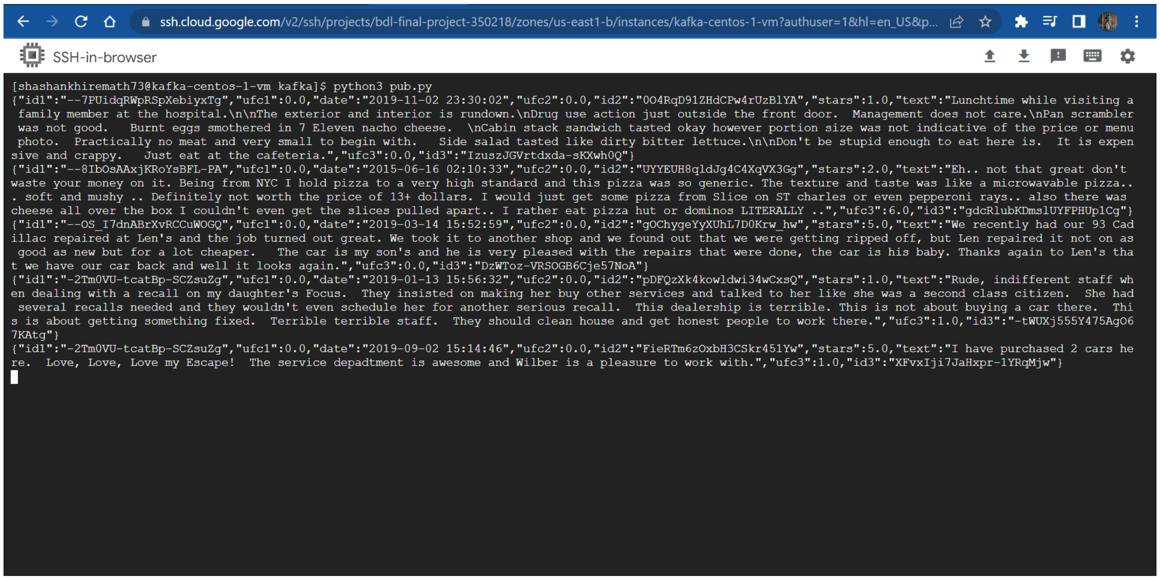
From the above two figures, we can see that the training set F1 score is 0.7165 and 0.6616 for the test set. This model took over 8 hours to train and evaluate. We could not try different hyperparameter values to improve accuracy before the report deadline.

6 Kafka Streaming

For the Kafka streaming task, we have created the publisher and subscriber. Code for these tasks can be found attached. The publisher and subscriber run simultaneously. The subscriber takes the data that is provided by the publisher. Upon receiving the data, the subscriber runs the model and predicts the output which can be seen in the log of the subscriber job.

6.1 Publisher

A kafka VM was created and the publisher code was run on the virtual machine SSH window. It is shown as follows:


ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&p...
SSH-in-browser

```
[shashankhiremath73@kafka-centos-1-vm kafka]$ python3 pub.py
{"id1":"--7PUidqRWpRSpXebiyxTg","ufc1":0.0,"date":"2019-11-02 23:30:02","ufc2":0.0,"id2":"0O4RqD91ZHdCPw4rUzBlYA","stars":1.0,"text":"Lunchtime while visiting a
 family member at the hospital.\n\nThe exterior and interior is rundown.\nDrug use action just outside the front door.  Management does not care.\nPan scrambler
 was not good.   Burnt eggs smothered in 7 Eleven nacho cheese.  \nCabin stack sandwich tasted okay however portion size was not indicative of the price or menu
 photo.  Practically no meat and very small to begin with.   Side salad tasted like dirty bitter lettuce.\n\nDon't be stupid enough to eat here is.  It is expen
sive and crappy.   Just eat at the cafeteria.","ufc3":0.0,"id3":"IzuszJGVrtdxda-sKXwh0Q"}
{"id1":"--8IbOsAAxjKRoYsBFL-PA","ufc1":0.0,"date":"2015-06-16 02:10:33","ufc2":0.0,"id2":"UYYEUH8qldJg4C4XqVX3Gg","stars":2.0,"text":"Eh.. not that great don't
waste your money on it. Being from NYC I hold pizza to a very high standard and this pizza was so generic. The texture and taste was like a microwavable pizza..
. soft and mushy .. Definitely not worth the price of 13+ dollars. I would just get some pizza from Slice on ST charles or even pepperoni rays.. also there was
cheese all over the box I couldn't even get the slices pulled apart.. I rather eat pizza hut or dominos LITERALLY ..","ufc3":6.0,"id3":"gdcRlubKDmslUYFPHUp1Cg"}
{"id1":"--OS_I7dnABrXvRCCuWOGQ","ufc1":0.0,"date":"2019-03-14 15:52:59","ufc2":0.0,"id2":"gOChygeYyXUhL7D0Krw_hw","stars":5.0,"text":"We recently had our 93 Cad
illac repaired at Len's and the job turned out great. We took it to another shop and we found out that we were getting ripped off, but Len repaired it not on as
 good as new but for a lot cheaper.   The car is my son's and he is very pleased with the repairs that were done, the car is his baby. Thanks again to Len's tha
t we have our car back and well it looks again.","ufc3":0.0,"id3":"DzWToz-VRSOGB6Cje57NoA"}
{"id1":"-2Tm0VU-tcatBp-SCZsuZg","ufc1":0.0,"date":"2019-01-13 15:56:32","ufc2":0.0,"id2":"pDFQzXk4kowldwi34wCxsQ","stars":1.0,"text":"Rude, indifferent staff wh
en dealing with a recall on my daughter's Focus.  They insisted on making her buy other services and talked to her like she was a second class citizen.  She had
 several recalls needed and they wouldn't even schedule her for another serious recall.  This dealership is terrible. This is not about buying a car there.  Thi
s is about getting something fixed.  Terrible terrible staff.  They should clean house and get honest people to work there.","ufc3":1.0,"id3":"-tWUXj555Y475AgO6
7KAtg"}
{"id1":"-2Tm0VU-tcatBp-SCZsuZg","ufc1":0.0,"date":"2019-09-02 15:14:46","ufc2":0.0,"id2":"FieRTm6zOxbH3CSkr451Yw","stars":5.0,"text":"I have purchased 2 cars he
re.  Love, Love, Love my Escape!  The service depadtment is awesome and Wilber is a pleasure to work with.","ufc3":1.0,"id3":"XFvxIji7JaHxpr-1YRqMjw"}
```

Figure 18: Publisher

We can see from the figure above that the output in the window corresponds to each data point and the output shows up in the JSON format.

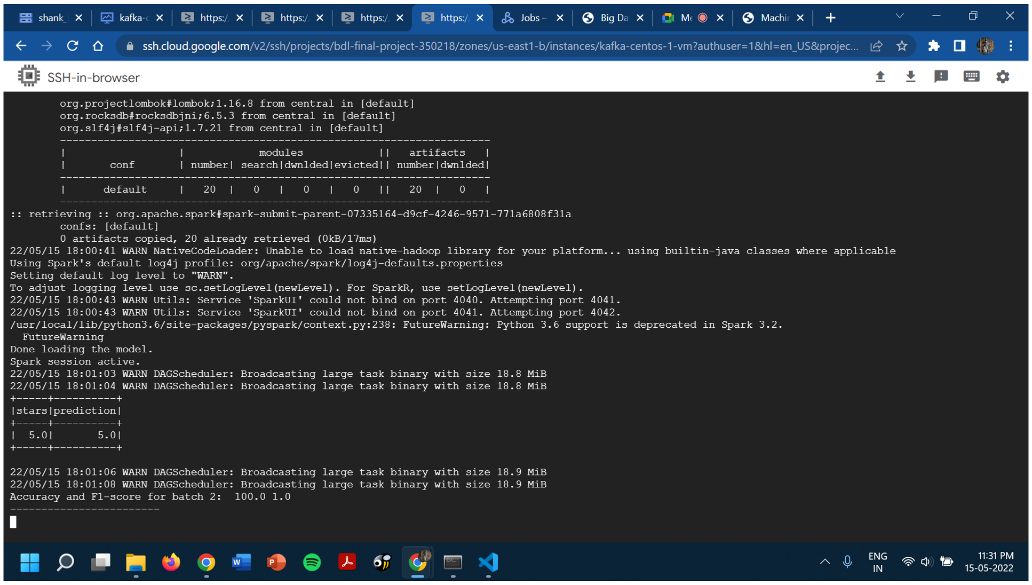
6.2 Subscriber

We establish a subscriber on a separate Dataproc cluster simultaneously. As the subscriber code runs, the specifics of the job can be found below:

The outputs are seen in batches. So as we update the publisher (or when it does while going over the dataset), each review is taken in one batch and the accuracy is calculated cumulatively over the tested reviews.

To test the real-time nature of this environment, we input our own review in the required 'json' format in the SSH window of the publisher. This input is then taken by the subscriber to run the model over and predict the stars rating.

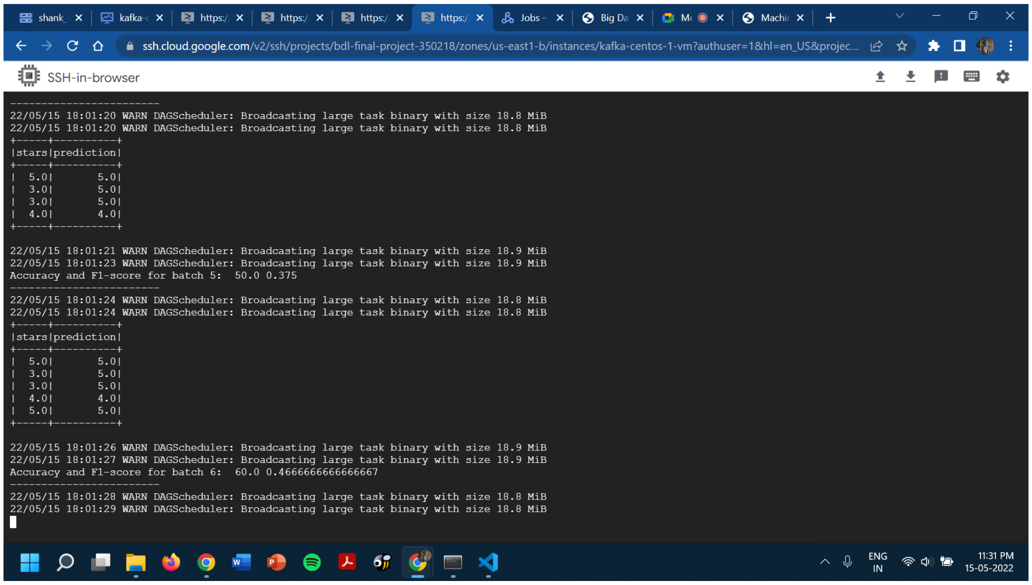
The results for some different batches are shown below:


shank_
kafka-c
https:/
https:/
https:/
https:/
Jobs –
Big Da
Me
Machi
ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&projec...
SSH-in-browser
ENG
IN
11:31 PM
15-05-2022

```
        org.projectlombok#lombok;1.16.8 from central in [default]
        org.rocksdb#rocksdbjni;6.5.3 from central in [default]
        org.slf4j#slf4j-api;1.7.21 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   20  |   0   |   0   |   0   ||   20  |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-07335164-d9cf-4246-9571-771a6808f31a
        confs: [default]
        0 artifacts copied, 20 already retrieved (0kB/17ms)
22/05/15 18:00:41 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
22/05/15 18:00:43 WARN Utils: Service 'SparkUI' could not bind on port 4040. Attempting port 4041.
22/05/15 18:00:43 WARN Utils: Service 'SparkUI' could not bind on port 4041. Attempting port 4042.
/usr/local/lib/python3.6/site-packages/pyspark/context.py:238: FutureWarning: Python 3.6 support is deprecated in Spark 3.2.
  FutureWarning
Done loading the model.
Spark session active.
22/05/15 18:01:03 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:04 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:06 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:08 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 2:  100.0 1.0
------------------------
█
```

Figure 19: Subscriber 1

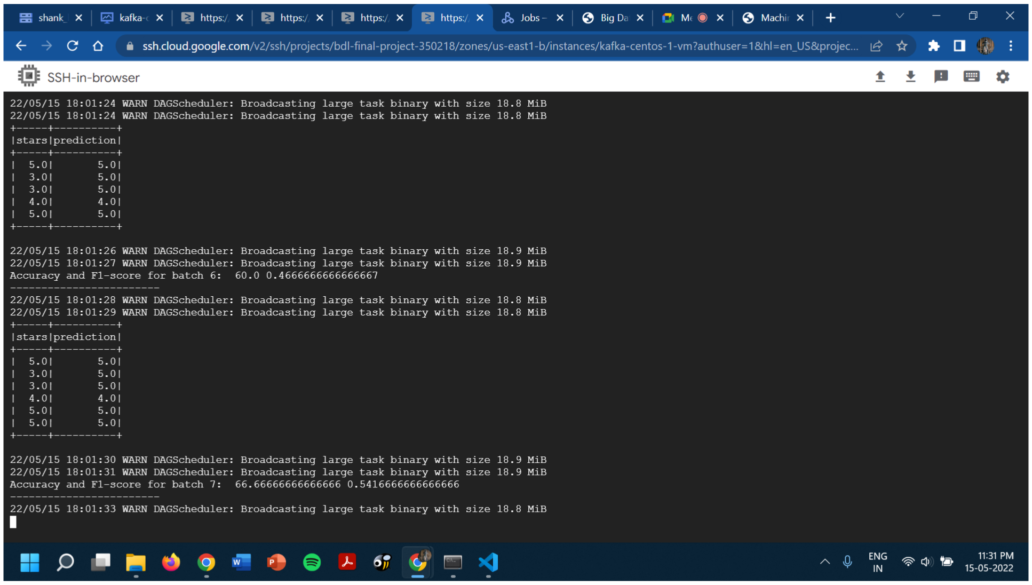

shank_
kafka-c
https:/
https:/
https:/
https:/
Jobs –
Big Da
Me
Machi
ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&projec...
SSH-in-browser
ENG
IN
11:31 PM
15-05-2022

```
------------------------
22/05/15 18:01:20 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:20 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
+-----+----------+

22/05/15 18:01:21 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:23 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 5:  50.0 0.375
------------------------
22/05/15 18:01:24 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:24 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:26 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:27 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 6:  60.0 0.4666666666666667
------------------------
22/05/15 18:01:28 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:29 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
█
```

Figure 20: Subscriber 2



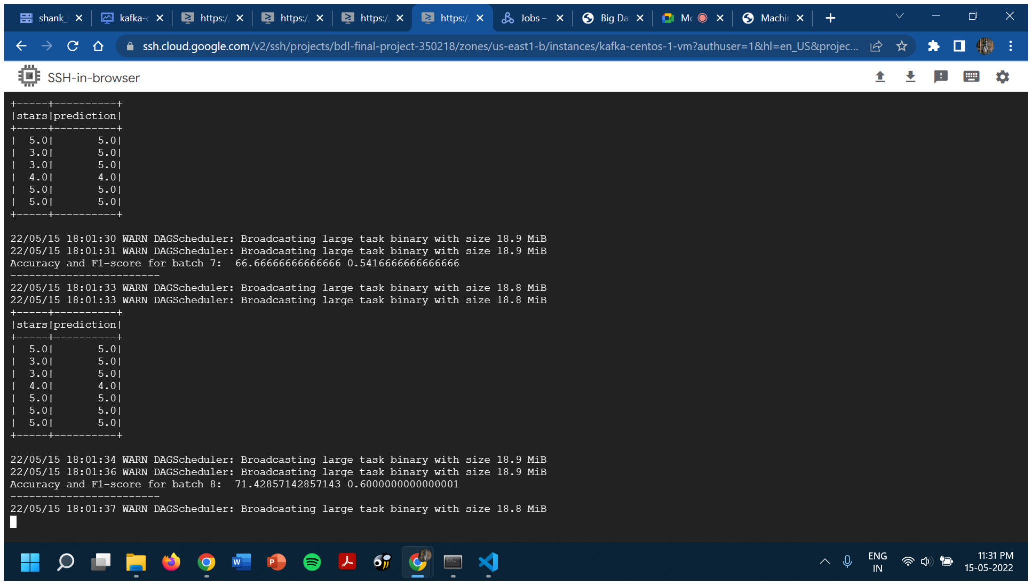
ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&projec...
SSH-in-browser

```
22/05/15 18:01:24 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:24 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:26 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:27 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 6:  60.0 0.4666666666666667
------------------------
22/05/15 18:01:28 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:29 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:30 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:31 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 7:  66.66666666666666 0.5416666666666666
------------------------
22/05/15 18:01:33 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
```

Figure 21: Subscriber 3



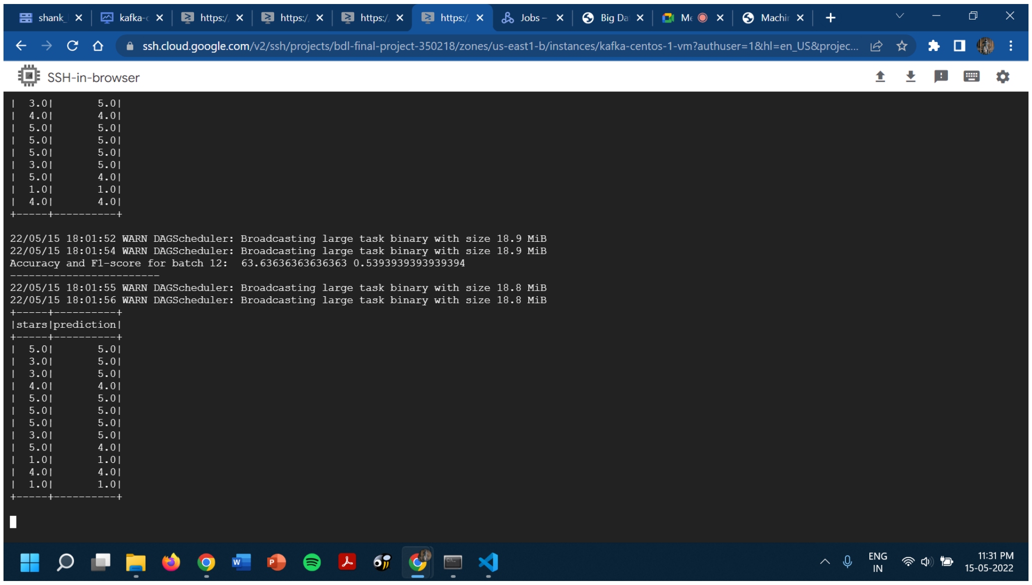
ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&projec...
SSH-in-browser

```
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:30 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:31 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 7:  66.66666666666666 0.5416666666666666
------------------------
22/05/15 18:01:33 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:33 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
|  5.0|       5.0|
|  5.0|       5.0|
+-----+----------+

22/05/15 18:01:34 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:36 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 8:  71.42857142857143 0.6000000000000001
------------------------
22/05/15 18:01:37 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
```

Figure 22: Subscriber 4



ssh.cloud.google.com/v2/ssh/projects/bdl-final-project-350218/zones/us-east1-b/instances/kafka-centos-1-vm?authuser=1&hl=en_US&projec...
SSH-in-browser

```
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
|  5.0|       5.0|
|  5.0|       5.0|
|  3.0|       5.0|
|  5.0|       4.0|
|  1.0|       1.0|
|  4.0|       4.0|
+-----+----------+

22/05/15 18:01:52 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
22/05/15 18:01:54 WARN DAGScheduler: Broadcasting large task binary with size 18.9 MiB
Accuracy and F1-score for batch 12:  63.63636363636363 0.5393939393939394
------------------------
22/05/15 18:01:55 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
22/05/15 18:01:56 WARN DAGScheduler: Broadcasting large task binary with size 18.8 MiB
+-----+----------+
|stars|prediction|
+-----+----------+
|  5.0|       5.0|
|  3.0|       5.0|
|  3.0|       5.0|
|  4.0|       4.0|
|  5.0|       5.0|
|  5.0|       5.0|
|  5.0|       5.0|
|  3.0|       5.0|
|  5.0|       4.0|
|  1.0|       1.0|
|  4.0|       4.0|
|  1.0|       1.0|
+-----+----------+
```

Figure 23: Subscriber 5

While testing the model on streaming data, the model was usually found to have an accuracy in the range 60-70% (with some outliers) and an F1 score of around 0.50-0.55.

7 Challenges

Some challenges we faced during the course of this project are enumerated below:

1. The provided CSV file contained commas (within the quotation marks) which were not to be inferred as delimiters. Furthermore, new lines were inserted in between quotes causing the data to be read incorrectly into the dataframe.

2. Few problems were faced in utilizing the spark NLP module in the codes. This eventually led a lot of time spent in identifying and utilizing the correct jar files and correct versions of pyspark, spark-nlp and kafka to be used. Hence, we had to use structured streaming and we ran the sub.py file on a new window of the kafka VM.

8 Conclusion

1. The given dataset of Yelp is highly skewed. The numerical variables have very less correlation with the 'stars' (output) variable.

2. The logistic regression model works the best for the given dataset.

3. The skewness of the model plays a major role and because of the large computation time, the accuracy is low for the given dataset. Since the 5 stars in the rating are not exactly orthogonal, we feel that this should be treated as a regression problem.

4. Kafka streaming was executed properly as shown above and we were able to obtain accuracies on the test data in real time.

5. The training set accuracy is 71.65% and the test set accuracy is 66.16%.