Please copy traning data from https://drive.google.com/file/d/1uNsbvMDz7Zz5cyskjNe0HL5LLXlpVvdX/view?usp=sha Markdown
container.

This data is used for model training.

In this notebook, we will use machine learning and apply NLP techniques to train a machine learning model. The model will use `Reviews` data to predict ratings

What we are going to do:

- Step 1: Prepare the training data for the machine learning training.
- Step 2: Train the machine learning model;
- Step 3: Save the model to a Azure storage folder so that you can use it for future prediction.

Creating the Reviews dataframe

✓ 11/1/2024 (< 1s) 3

```
#Original Statement
#spark._jsc.hadoopConfiguration().set("fs.azure.account.key.storagewcdb11cc.dfs.core.windows.net", "8/MpKW/eANtthZIzqYxK574mC/wr2/
yQ7LDk0Um/4mxyQY8F92TdeclAQOxvvQjdaOoHgdnQ63yP+AStC6chpA==")
```

Create the Databricks mountpoint

5

```
storageAccountName = 'storagewcdb11cc'
containerName = 'project'
applicationId = 'e9f1f100-81d7-401e-aaf5-7249fdd0fd7a'
directoryID = '7cb3acb0-5084-47dc-b739-e1160792f61c'
secretValue = 'p1m8Q~cAdgSDxddK9zfqpZaopaT5rN-6WoWXDcUk'
endpoint = 'https://login.microsoftonline.com/' + directoryID + '/oauth2/token'
source = 'abfss://' + containerName + '@' + storageAccountName + '.dfs.core.windows.net/'
mountPoint = "/mnt/deBDproject"

configs = {"fs.azure.account.auth.type": "OAuth",
           "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
           "fs.azure.account.oauth2.client.id": applicationId,
           "fs.azure.account.oauth2.client.secret": secretValue,
           "fs.azure.account.oauth2.client.endpoint": endpoint}

dbutils.fs.mount(source = source,mount_point = mountPoint, extra_configs = configs)
```

✓ 11/1/2024 (10s) 6

```
#Original Statement
#file_location = "abfss://project@storagewcdb11cc.dfs.core.windows.net/*"
file_location = "/mnt/deBDproject/de-yelp-train/*"

reviews = spark.read \
  .parquet(file_location)

#reviews = spark.read \
#  .option("basePath", file_location) \
#  .parquet(file_location + "*")
```

(1) Spark Jobs

reviews: pyspark.sql.dataframe.DataFrame

```
business_id: string
cool: long
date: string
funny: long
review_id: string
stars: double
text: string
useful: long
user_id: string
```

Unmounted subdirectory no longer used

8

```
#mountPoint = "/mnt/deBDproject"
mountPoint = "dbfs:/mnt/bd-project/"

dbutils.fs.unmount(mount_point = mountPoint)
```

Display file directory structure storing the model

✓ 11/2/2024 (11s) 10

```
display(
  dbutils.fs.ls("/mnt/deBDproject/de-yelp-train")
)
```

(3) Spark Jobs

Table +

	path	name
1	dbfs:/mnt/deBDproject/de-yelp-train/model/	model/
2	dbfs:/mnt/deBDproject/de-yelp-train/model_bak/	model_bak/
3	dbfs:/mnt/deBDproject/de-yelp-train/part-00000-tid-1068847200739902294-534edb4d-3b0a-4d29-81e5-7c1612f38f24-204-1-c000.snap...	part-00000-tid-1068847200739902294-53
4	dbfs:/mnt/deBDproject/de-yelp-train/part-00001-tid-1068847200739902294-534edb4d-3b0a-4d29-81e5-7c1612f38f24-205-1-c000.snap...	part-00001-tid-1068847200739902294-53
5	dbfs:/mnt/deBDproject/de-yelp-train/part-00002-tid-1068847200739902294-534edb4d-3b0a-4d29-81e5-7c1612f38f24-202-1-c000.snap...	part-00002-tid-1068847200739902294-53
6	dbfs:/mnt/deBDproject/de-yelp-train/part-00003-tid-1068847200739902294-534edb4d-3b0a-4d29-81e5-7c1612f38f24-203-1-c000.snap...	part-00003-tid-1068847200739902294-53
7	dbfs:/mnt/deBDproject/de-yelp-train/results/	results/
8	dbfs:/mnt/deBDproject/de-yelp-train/stringindexer/	stringindexer/
9	dbfs:/mnt/deBDproject/de-yelp-train/stringindexer_bak/	stringindexer_bak/

9 rows | 1.51 seconds runtime Refreshed 7 days ago

Display list of review subdirectories in "/mnt/deBDproject/de-yelp-daily/reviews"

11/2/2024 (1s) 12

```
display(
  dbutils.fs.ls("/mnt/deBDproject/de-yelp-daily/reviews")
)
```

(3) Spark Jobs

Table

	path	name	size	modificationTime
1	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-09-2...	dt=2024-09-21/	0	1730342254000
2	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-0...	dt=2024-10-05/	0	1730342253000
3	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-0...	dt=2024-10-08/	0	1730342254000
4	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-10/	0	1730342254000
5	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-13/	0	1730342254000
6	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-15/	0	1730342255000
7	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-16/	0	1730342255000
8	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-17/	0	1730342255000
9	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-1...	dt=2024-10-19/	0	1730342255000
10	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-2...	dt=2024-10-22/	0	1730342255000
11	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-2...	dt=2024-10-24/	0	1730342255000
12	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-2...	dt=2024-10-28/	0	1730342255000
13	dbfs:/mnt/deBDproject/de-yelp-daily/reviews/dt=2024-10-2...	dt=2024-10-29/	0	1730478573000

13 rows | 0.74 seconds runtime

Refreshed 7 days ago

11/2/2024 (1s) 13

```
%fs head /mnt/deBDproject/de-yelp-train/part-00000-tid-1068847200739902294-534edb4d-3b0a-4d29-81e5-7c1612f38f24-204-1-c000.snappy(1).parquet
```

```
[Truncated to first 65536 bytes]
<ş��:dORL4JE6tz3rJxVqkdKfegATRCy4M2ND4YK0uRbodV_v8gPVN2CJfXX6ooJt-Nc3vKt9TGInRkBWvuyJCjFVHYdzkPQ4TM4kHDHNzftSUtgpgyLlZ8wPxlzYL_Dwm7RdQzhC0FQgkP
qC63cwAI-baLmbEdxbR3F�Pw520vYWEpEYBf90rSr9NVPgmjsEdUsKpj9Xxu6pdjH0   
```

Display the results schema

✓ 11/2/2024 (<1s) 18

```
reviews.printSchema()
```

```
root
 |-- business_id: string (nullable = true)
 |-- cool: long (nullable = true)
 |-- date: string (nullable = true)
 |-- funny: long (nullable = true)
 |-- review_id: string (nullable = true)
 |-- stars: double (nullable = true)
 |-- text: string (nullable = true)
 |-- useful: long (nullable = true)
 |-- user_id: string (nullable = true)
```

Display the columns we will be focusing on. text = yelp reviews. stars = ratings

✓ 11/2/2024 (<1s) 20

```
reviews = reviews.select("text", "stars")
```

```
reviews: pyspark.sql.dataframe.DataFrame
    text: string
    stars: double
```

Save dataframe to memory for improved performance with repetitive use. Count the number of reviews.

✓ 11/2/2024 (7s) 22

```
# Saving the dataframe to memory for repetitive use
reviews.cache()
reviews.count()
```

▸ (3) Spark Jobs

```
698757
```

A partition is composed of a subset of rows in a table that share the same value for a predefined subset of columns called the partitioning columns. Using partitions can speed up queries against the table as well as data manipulation.

✓ 11/2/2024 (<1s) 24

```
print("Number of partitions before:", reviews.rdd.getNumPartitions())
# reviews = reviews.repartition(16)  # Adjust the number based on your cluster size and data
# print("Number of partitions after:", reviews.rdd.getNumPartitions())
```

```
Number of partitions before: 4
```

EDA

Exploratory Data Analysis

Load library to detect languages in text column

▶ ✓ 11/2/2024 (9s) 27

```
%pip install langid
```

```
Collecting langid
  Downloading langid-1.1.6.tar.gz (1.9 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.0/1.9 MB ? eta -:--:--
     ━━━━━━━━ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0.5/1.9 MB 13.6 MB/s eta 0:00:01
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 29.8 MB/s eta 0:00:00
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: numpy in /databricks/python3/lib/python3.11/site-packages (from langid) (1.23.5)
Building wheels for collected packages: langid
  Building wheel for langid (setup.py): started
  Building wheel for langid (setup.py): finished with status 'done'
  Created wheel for langid: filename=langid-1.1.6-py3-none-any.whl size=1941173 sha256=4daebc827146aae7c152917b0cffeafa35a8445534b2b136fb60603645
665b25
  Stored in directory: /root/.cache/pip/wheels/32/6a/b6/b7eb43a6ad55b139c15c5daa29f3707659cfa6944d3c696f5b
Successfully built langid
Installing collected packages: langid
Successfully installed langid-1.1.6
Note: you may need to restart the kernel using %restart_python or dbutils.library.restartPython() to use updated packages.
```

💡1

Import udf, StringType libraries Use UDF to detect the type of languages in the text (aka reviews) column

▶ ✓ 11/2/2024 (<1s) 29

```
import langid
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

# UDF to detect language
def detect_language(text):
    return langid.classify(text)[0]

detect_language_udf = udf(detect_language, StringType())

# Assuming df is your DataFrame and 'text' is the column with text data
reviews = reviews.withColumn("language", detect_language_udf(reviews.text))
```

▾ reviews: pyspark.sql.dataframe.DataFrame
  text: string
  stars: double
  language: string

Display the reviews dataframe. Scroll to the right to the [language] column to see what language it's written in.

11/2/2024 (15s) 31

```
display(reviews)
```

(1) Spark Jobs

Table | Visualization 1 +

	text	1.2 stars	language
1	I remember staying here from some business meetings about 20 years ago and was Impressed with their indoor conservatory. Well, fas...	4	en
2	Convenient location right next to my favorite nail shop. I come every time I get my nails done, which is often! It gets super busy here s...	4	en
3	It's huge- you don't realize you big from the outside, but this place never ends and every booth is different. The prices are usually prett...	5	en
4	I'm not sure what happened here, but this place went downhill fast. I just had a pizza steak and chicken parm sandwich that were a ste...	1	en
5	Finally, finally a "real pizzeria"!!! Food is fresh! And absolutely delicious. Service is great- so friendly and helpful. What a find!	5	en
6	This is a great place to watch a game, eat good food, or just grab a drink with a friend. Staff is fantastic, and prices are reasonable.	5	en
7	If you're looking for atmosphere, go somewhere else. If you're looking for giant saucy chicken wings (for a good price) and great NY St...	4	en
8	The food is definitely authentic. Even the "mild" version is really, really hot... so be prepared. But regardless, their curry dish was excelle...	3	en
9	LOVED! We had Michael as our guide, and he was truly the best. He was very hands on with the tour, showing us places to take picture...	5	en
10	The food is not the "greasy-spoon" diner experience I was hoping for. Too many "unique" menu options and too centered around sout...	3	en
11	What can I say, the food here was beyond delicious. My husband and I were looking for a place to have lunch and just so happened to ...	5	en
12	This was an amazing event space with exceptional attention to detail for each and every member of our team! We held a corporate retr...	5	en
13	I had the Mexican tostadas and rice and beans and the food was very good!! I had the chicken on the tostados and it was good. Definie...	5	en
14	very cute little burger shop. Figured I should try it out. I like the hippie vibe. The burger itself, however, was a bit underseasoned and m...	3	en
15	Novio and I wanted burgers but in the miniature variety so DDSB was the perfect choice. Order at the counter, got 2 double cheeseburr...	5	en

3,559+ rows | Truncated data due to byte limit | 15.13 seconds runtime Refreshed 7 days ago

Cache the reviews dataframe again due to dataframe changes

11/2/2024 (4m) 33

```
reviews.cache()
reviews.count()
```

(3) Spark Jobs

```
698757
```

Group by the [language] column on the dataframe to see what languages there are in the text (aka reviews) column. Language is mostly english, but you can see there are a lot of different languages used in reviews.

11/2/2024 (1s) 35

```
language_counts = reviews.groupBy("language").count()
language_counts.show()
```

(2) Spark Jobs

language_counts: pyspark.sql.dataframe.DataFrame = [language: string, count: long]

```
|      ro|     7|
|      pl|     4|
|      pt|    29|
|      oc|     3|
|      gl|     1|
|      ms|     5|
|      cs|     2|
|      de|    99|
|      br|     5|
|      es|   276|
|      it|    72|
|      af|    13|
|      sv|    13|
|      nl|    25|
|      lt|     1|
|      no|     6|
|      zh|    52|
|      fr|    78|
+--------+------+
only showing top 20 rows
```

Filter the reviews dataframe to only include the english language

✓ 11/2/2024 (<1s) 38

```
reviews = reviews.filter(reviews.language == "en")
reviews = reviews.filter(reviews.language.isNotNull())
```

```
reviews: pyspark.sql.dataframe.DataFrame
    text: string
    stars: double
    language: string
```

cache the reviews dataframe to data changes

✓ 11/2/2024 (2s) 40

```
reviews.cache()
reviews.count()
```

▸ (3) Spark Jobs

```
697986
```

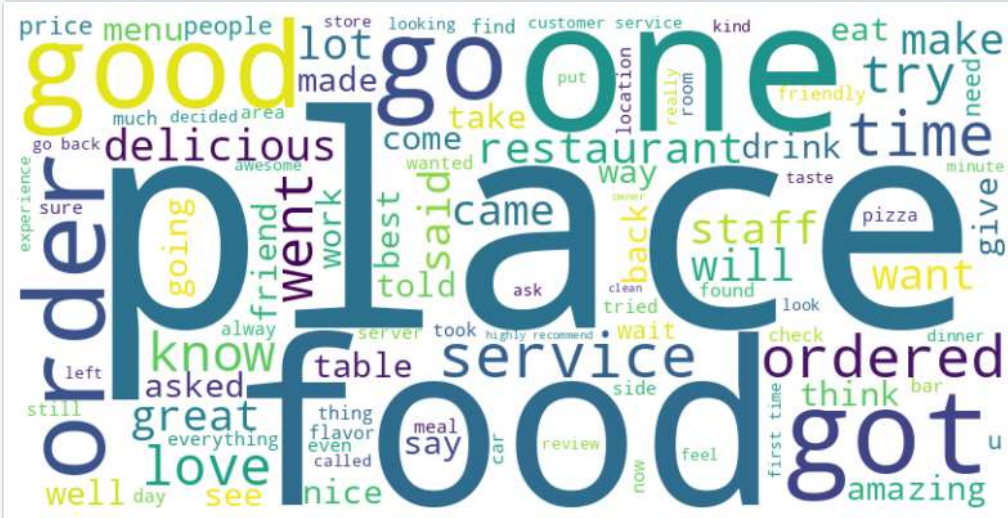
Plotting Words using WordCloud

Collect texts from PySpark DataFrame to the driver. Needed to increase compute cluster to increase memory to support the WordCloud library.

✓ 11/2/2024 (3m) 42

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt

# Collect texts from PySpark DataFrame to the driver
all_texts = reviews.select("text").rdd.flatMap(lambda x: x).collect()

# Join all text items into a single string
all_text = ' '.join(all_texts)

# Generate WordCloud
wordcloud = WordCloud(width=800, height=400, background_color='white', max_words=100).generate(all_text)

# Display the WordCloud using matplotlib
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

▸ (1) Spark Jobs



Text Normalization

Normalization of text column. Removing unwanted characters and symbols from text (aka results) column. Set the case to lowercase.

✓ 11/2/2024 (<1s) 45

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import lower, regexp_replace, trim
from pyspark.ml.feature import Tokenizer, StopWordsRemover

reviews = reviews.withColumn("text", lower(reviews.text)) # Lowercase Conversion
reviews = reviews.withColumn("text", regexp_replace("text", "[^a-zA-Z\\s]", "")) # Remove Punctuation
reviews = reviews.withColumn("text", regexp_replace("text", "\\s+", " ")) # Remove Extra Spaces
reviews = reviews.withColumn("text", trim(reviews.text))
```

reviews: pyspark.sql.dataframe.DataFrame = [text: string, stars: double ... 1 more field]

✓ 11/2/2024 (1s) 46

```
display(reviews)
```

(1) Spark Jobs

Table

	text	1.2 stars	language
1	i remember staying here from some business meetings about years ago and was impressed with their indoor conservatory well fast for...	4	en
2	convenient location right next to my favorite nail shop i come every time i get my nails done which is often it gets super busy here so ...	4	en
3	its huge you dont realize you big from the outside but this place never ends and every booth is different the prices are usually pretty re...	5	en
4	im not sure what happened here but this place went downhill fast i just had a pizza steak and chicken parm sandwich that were a step ...	1	en
5	finally finally a real pizzeria food is fresh and absolutely delicious service is great so friendly and helpful what a find	5	en
6	this is a great place to watch a game eat good food or just grab a drink with a friend staff is fantastic and prices are reasonable	5	en
7	if youre looking for atmosphere go somewhere else if youre looking for giant saucy chicken wings for a good price and great ny style ...	4	en
8	the food is definitely authentic even the mild version is really really hot so be prepared but regardless their curry dish was excellent sou...	3	en
9	loved we had michael as our guide and he was truly the best he was very hands on with the tour showing us places to take pictures an...	5	en
10	the food is not the greasyspoon diner experience i was hoping for too many unique menu options and too centered around southwest...	3	en
11	what can i say the food here was beyond delicious my husband and i were looking for a place to have lunch and just so happened to st...	5	en
12	this was an amazing event space with exceptional attention to detail for each and every member of our team we held a corporate retre...	5	en
13	i had the mexican tostadas and rice and beans and the food was very good i had the chicken on the tostados and it was good definietl...	5	en
14	very cute little burger shop figured i should try it out i like the hippie vibe the burger itself however was a bit underseasoned and mush...	3	en
15	novio and i wanted burgers but in the miniature variety so ddsb was the perfect choice order at the counter got double cheeseburgers ...	5	en

3,749+ rows | Truncated data due to byte limit | 0.99 seconds runtime Refreshed 7 days ago

Tokenization

Tokenization is the process of breaking text into individual words, phrases, symbols, or other meaningful elements called tokens. This step is using simple Word Tokenization. Types of Tokenization Word Tokenization: Breaking text into individual words. Sentence Tokenization: Breaking text into individual sentences. Stemming: Reducing words to their base or root form. For example, "running" → "run". Lemmatization: Reducing words to their base form using a lexical knowledge base. It's more accurate than stemming. For example, "better" → "good". Stop Word Removal: Removing common words that may not add significant meaning in analysis, like "and", "the", "is".

✓ 11/2/2024 (1s) 48

```
from pyspark.sql.functions import regexp_replace, trim, split, translate
from pyspark.ml.feature import Tokenizer

tokenizer = Tokenizer(inputCol="text", outputCol="tokens")
tokenized = tokenizer.transform(reviews)
```

tokenized: pyspark.sql.dataframe.DataFrame = [text: string, stars: double ... 2 more fields]

11/2/2024 (2s) 49

```
display(tokenized)
```

(1) Spark Jobs

Table +

	text	1.2 stars	language
1	i remember staying here from some business meetings about years ago and was impressed with their indoor conservatory well fast for...	4	en
2	convenient location right next to my favorite nail shop i come every time i get my nails done which is often it gets super busy here so ...	4	en
3	its huge you dont realize you big from the outside but this place never ends and every booth is different the prices are usually pretty re...	5	en
4	im not sure what happened here but this place went downhill fast i just had a pizza steak and chicken parm sandwich that were a step ...	1	en
5	finally finally a real pizzeria food is fresh and absolutely delicious service is great so friendly and helpful what a find	5	en
6	this is a great place to watch a game eat good food or just grab a drink with a friend staff is fantastic and prices are reasonable	5	en
7	if youre looking for atmosphere go somewhere else if youre looking for giant saucy chicken wings for a good price and great ny style ...	4	en
8	the food is definitely authentic even the mild version is really really hot so be prepared but regardless their curry dish was excellent sou...	3	en
9	loved we had michael as our guide and he was truly the best he was very hands on with the tour showing us places to take pictures an...	5	en
10	the food is not the greasyspoon diner experience i was hoping for too many unique menu options and too centered around southwest...	3	en
11	what can i say the food here was beyond delicious my husband and i were looking for a place to have lunch and just so happened to st...	5	en
12	this was an amazing event space with exceptional attention to detail for each and every member of our team we held a corporate retre...	5	en
13	i had the mexican tostadas and rice and beans and the food was very good i had the chicken on the tostados and it was good definietl...	5	en
14	very cute little burger shop figured i should try it out i like the hippie vibe the burger itself however was a bit underseasoned and mush...	3	en
15	novio and i wanted burgers but in the miniature variety so ddsb was the perfect choice order at the counter got double cheeseburgers ...	5	en

1,597+ rows | Truncated data due to byte limit | 1.75 seconds runtime Refreshed 7 days ago

Create a new column [filtered] with some words removed. "I" "and" etc...

11/2/2024 (<1s) 51

```
from pyspark.ml.feature import StopWordsRemover

stopword_remover = StopWordsRemover(inputCol="tokens", outputCol="filtered")
stopword = stopword_remover.transform(tokenized)
```

```
stopword: pyspark.sql.dataframe.DataFrame
  text: string
  stars: double
  language: string
  tokens: array
    element: string
  filtered: array
    element: string
```

✓ 11/2/2024 (2s) 52

```
display(tokenized)
```

▸ (1) Spark Jobs

Table +

	text	1.2 stars	language
1	i remember staying here from some business meetings about years ago and was impressed with their indoor conservatory well fast for...	4	en
2	convenient location right next to my favorite nail shop i come every time i get my nails done which is often it gets super busy here so ...	4	en
3	its huge you dont realize you big from the outside but this place never ends and every booth is different the prices are usually pretty re...	5	en
4	im not sure what happened here but this place went downhill fast i just had a pizza steak and chicken parm sandwich that were a step ...	1	en
5	finally finally a real pizzeria food is fresh and absolutely delicious service is great so friendly and helpful what a find	5	en
6	this is a great place to watch a game eat good food or just grab a drink with a friend staff is fantastic and prices are reasonable	5	en
7	if youre looking for atmosphere go somewhere else if youre looking for giant saucy chicken wings for a good price and great ny style ...	4	en
8	the food is definitely authentic even the mild version is really really hot so be prepared but regardless their curry dish was excellent sou...	3	en
9	loved we had michael as our guide and he was truly the best he was very hands on with the tour showing us places to take pictures an...	5	en
10	the food is not the greasyspoon diner experience i was hoping for too many unique menu options and too centered around southwest...	3	en
11	what can i say the food here was beyond delicious my husband and i were looking for a place to have lunch and just so happened to st...	5	en
12	this was an amazing event space with exceptional attention to detail for each and every member of our team we held a corporate retre...	5	en
13	i had the mexican tostadas and rice and beans and the food was very good i had the chicken on the tostados and it was good definietl...	5	en
14	very cute little burger shop figured i should try it out i like the hippie vibe the burger itself however was a bit underseasoned and mush...	3	en
15	novio and i wanted burgers but in the miniature variety so ddsb was the perfect choice order at the counter got double cheeseburgers ...	5	en

1,597+ rows | Truncated data due to byte limit | 1.52 seconds runtime Refreshed 7 days ago

Extracts a vocabulary from document collections and generates a CountVectorizerModel.
is a tool used for converting a collection of text documents into a matrix of token counts. In other words, it transforms raw text data into numerical data, which can be used by machine learning algorithms.

Here's what the CountVectorizer does step by step:

```
Tokenization: It splits each text document into individual words or tokens. Typically, this involves removing punctuation and p
Vocabulary Building: The CountVectorizer builds a vocabulary of all unique words that appear across the entire collection of do
Feature Extraction: For each document, it counts how many times each word (or token) from the vocabulary appears in the document
Sparse Matrix Representation: The result is usually a sparse matrix because most text data is sparse in terms of word occurrenc
```

Example:

Let's say you have a dataset with three documents:

```
"I love machine learning"
"Machine learning is fun"
"I love programming"
```

The CountVectorizer might build a vocabulary like this:

```
"I", "love", "machine", "learning", "is", "fun", "programming"
```

Then, it would generate a matrix like this (each row corresponds to a document, and each column corresponds to a word from the vocabulary):
Document I love machine learning is fun programming 1 1 1 1 1 0 0 0 2 0 0 1 1 1 1 0 3 1 1 0 0 0 0 1

Use Cases: Text Classification: For tasks like spam detection, sentiment analysis, or topic modeling. Feature Engineering: Converting text data into a form that can be fed into machine learning models. Text Analysis: Exploring the frequency of words in different documents.

✓ 11/2/2024 (30s) 54

```
from pyspark.ml.feature import CountVectorizer

cv = CountVectorizer(vocabSize=2**16, inputCol="filtered", outputCol='cv')
cv_model = cv.fit(stopword)
text_cv = cv_model.transform(stopword)
```

▸ (2) Spark Jobs

▾ (1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

▸ text_cv: pyspark.sql.dataframe.DataFrame

1

Scroll to the right for the cv column displaying the Count Vectorized Model

✓ 11/2/2024 (3s) 56

```
display(text_cv)
```

▸ (1) Spark Jobs

Table +

	text	1.2 stars	language
1	i remember staying here from some business meetings about years ago and was impressed with their indoor conservatory well fast for...	4	en
2	convenient location right next to my favorite nail shop i come every time i get my nails done which is often it gets super busy here so ...	4	en
3	its huge you dont realize you big from the outside but this place never ends and every booth is different the prices are usually pretty re...	5	en
4	im not sure what happened here but this place went downhill fast i just had a pizza steak and chicken parm sandwich that were a step ...	1	en
5	finally finally a real pizzeria food is fresh and absolutely delicious service is great so friendly and helpful what a find	5	en
6	this is a great place to watch a game eat good food or just grab a drink with a friend staff is fantastic and prices are reasonable	5	en
7	if youre looking for atmosphere go somewhere else if youre looking for giant saucy chicken wings for a good price and great ny style ...	4	en
8	the food is definitely authentic even the mild version is really really hot so be prepared but regardless their curry dish was excellent sou...	3	en
9	loved we had michael as our guide and he was truly the best he was very hands on with the tour showing us places to take pictures an...	5	en
10	the food is not the greasyspoon diner experience i was hoping for too many unique menu options and too centered around southwest...	3	en
11	what can i say the food here was beyond delicious my husband and i were looking for a place to have lunch and just so happened to st...	5	en
12	this was an amazing event space with exceptional attention to detail for each and every member of our team we held a corporate retre...	5	en
13	i had the mexican tostadas and rice and beans and the food was very good i had the chicken on the tostados and it was good definietl...	5	en
14	very cute little burger shop figured i should try it out i like the hippie vibe the burger itself however was a bit underseasoned and mush...	3	en
15	novio and i wanted burgers but in the miniature variety so ddsb was the perfect choice order at the counter got double cheeseburgers ...	5	en

965+ rows | Truncated data due to byte limit | 2.66 seconds runtime Refreshed 7 days ago

Calculating Inverse Document Frequency on the cv_text column. Converting tokens or words into numerical values so that they can be fed into machine learning models. Methods One-Hot Encoding: Each word is represented as a vector with a '1' in its corresponding position in the vocabulary and '0' elsewhere. Word Embeddings: Dense vector representations where similar words have similar vectors. Examples include Word2Vec, GloVe, and FastText. TF-IDF (Term Frequency-Inverse Document Frequency): Weighs words based on their importance in a document relative to a collection of documents.

✓ 11/2/2024 (30s) 59

```
from pyspark.ml.feature import IDF

idf = IDF(inputCol='cv', outputCol="features", minDocFreq=5) #minDocFreq: remove sparse terms
idf_model = idf.fit(text_cv)
text_idf = idf_model.transform(text_cv)
```

▸ (1) Spark Jobs

▾ (1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

▸ text_idf: pyspark.sql.dataframe.DataFrame

StringIndexer - Takes [stars] column, creates label column and creates index.

In Databricks, the StringIndexer is a feature of the Spark MLlib library that is used for converting categorical string values into numerical indices. It's an important preprocessing step when you need to work with machine learning algorithms that require numerical input, but your data contains categorical variables (e.g., "red", "blue", "green"). What StringIndexer Does:

```
Converts Categorical Strings to Numeric Indices:
    The StringIndexer takes a column of string labels and converts them into numeric values. For example, if you have a column

Ordering Based on Frequency:
    By default, the StringIndexer assigns numeric indices based on the frequency of the categories in the dataset. The most fre

Handles New Categories:
    In case of new categories appearing in new data (that weren't seen during training), StringIndexer can handle them by using
```

Example Use Case:

Assume you have a column Color with values: "Red", "Green", "Blue", "Red", "Blue".

```
Input (before StringIndexer):
```

Color

Red Green Blue Red Blue

Output (after applying StringIndexer):

```
Color  | Index
---------------
Red    | 0
Green  | 1
Blue   | 2
Red    | 0
Blue   | 2
```

Key Parameters:

```
inputCol: The name of the input column (the string column you want to index).
outputCol: The name of the output column that will store the numeric indices.
handleInvalid: Option to handle invalid input during transformation. You can set this to 'skip' (skip rows with invalid labels)
```

✓ 11/2/2024 (3s) 61

```
from pyspark.ml.feature import StringIndexer

label_encoder = StringIndexer(inputCol = "stars", outputCol = "label")
le_model = label_encoder.fit(text_idf)
final = le_model.transform(text_idf)
```

▸ (2) Spark Jobs

▾ (1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

▸ final: pyspark.sql.dataframe.DataFrame

Scroll to the right to see [features] index

✓ 11/2/2024 (3s) 63

```
display(final)
```

▸ (1) Spark Jobs

Table +

	text	1.2 stars	language
1	i remember staying here from some business meetings about years ago and was impressed with their indoor conservatory well fast for...	4	en
2	convenient location right next to my favorite nail shop i come every time i get my nails done which is often it gets super busy here so ...	4	en
3	its huge you dont realize you big from the outside but this place never ends and every booth is different the prices are usually pretty re...	5	en
4	im not sure what happened here but this place went downhill fast i just had a pizza steak and chicken parm sandwich that were a step ...	1	en
5	finally finally a real pizzeria food is fresh and absolutely delicious service is great so friendly and helpful what a find	5	en
6	this is a great place to watch a game eat good food or just grab a drink with a friend staff is fantastic and prices are reasonable	5	en
7	if youre looking for atmosphere go somewhere else if youre looking for giant saucy chicken wings for a good price and great ny style ...	4	en
8	the food is definitely authentic even the mild version is really really hot so be prepared but regardless their curry dish was excellent sou...	3	en
9	loved we had michael as our guide and he was truly the best he was very hands on with the tour showing us places to take pictures an...	5	en
10	the food is not the greasyspoon diner experience i was hoping for too many unique menu options and too centered around southwest...	3	en
11	what can i say the food here was beyond delicious my husband and i were looking for a place to have lunch and just so happened to st...	5	en
12	this was an amazing event space with exceptional attention to detail for each and every member of our team we held a corporate retre...	5	en
13	i had the mexican tostadas and rice and beans and the food was very good i had the chicken on the tostados and it was good definietl...	5	en
14	very cute little burger shop figured i should try it out i like the hippie vibe the burger itself however was a bit underseasoned and mush...	3	en
15	novio and i wanted burgers but in the miniature variety so ddsb was the perfect choice order at the counter got double cheeseburgers ...	5	en

662+ rows | Truncated data due to byte limit | 2.56 seconds runtime Refreshed 7 days ago

Split data into 80% 20%. 80% to train model, 20% to test data

✓ 11/2/2024 (<1s) 65

```
train_data, test_data = final.randomSplit([0.8, 0.2], seed=2024)
```

▸ train_data: pyspark.sql.dataframe.DataFrame
▸ test_data: pyspark.sql.dataframe.DataFrame

cache data for data processing model

✓ 11/2/2024 (30s) 67

```
train_data.cache()
train_data.count()
```

▸ (3) Spark Jobs

```
558142
```

smaller/shallow decision trees. maxDepth = 5 levels deep. Seed makes it more deterministic. Train and then test model. Transform predicts labels. Creates model, predicts labels. Assigns to predictions dataframe.

✓ 11/2/2024 (10m) 69

```
from pyspark.ml.classification import RandomForestClassifier


rf = RandomForestClassifier(numTrees=100, maxDepth=5, seed=42)
rf_model = rf.fit(train_data)
predictions = rf_model.transform(test_data)
```

▸ (15) Spark Jobs

▾ (1) MLflow run
Logged 1 run to an experiment in MLflow. Learn more

▸ predictions: pyspark.sql.dataframe.DataFrame

Stale widget: cannot display widget because the python repl changed. Please rerun the notebook

Stale widget: cannot display widget because the python repl changed. Please rerun the notebook

Model Evalution

Overview: MulticlassClassificationEvaluator in Databricks (or Spark MLlib) is an essential tool for assessing the performance of multiclass classification models. It calculates various metrics such as accuracy, F1 score, precision, and recall, helping you understand how well your model performs on unseen data.

The MulticlassClassificationEvaluator in Databricks (specifically in Apache Spark MLlib) is a utility used for evaluating the performance of machine learning classification models, particularly for multiclass classification problems.

It computes evaluation metrics based on the predictions made by a classification model, such as accuracy, and can be used to assess how well the model is performing in terms of correctly predicting the class labels. Key Points about MulticlassClassificationEvaluator:

```
Purpose: It is specifically designed for classification tasks where the model predicts one of several possible classes. It suppo
Metrics:
    Accuracy: The most commonly used evaluation metric, which calculates the proportion of correct predictions out of the total
    It can also calculate other metrics like weightedPrecision, weightedRecall, and weightedF1Score, but accuracy is the most co
Inputs:
    It requires a predicted label column (the column with the predicted class labels from the model).
    It also requires the true label column (the actual class labels from the test data).
```

How It Works: The MulticlassClassificationEvaluator takes the model's predictions and compares them to the actual values in the test set, and then computes the evaluation metrics. By default, it calculates accuracy (i.e., the percentage of correct predictions), but you can specify different metrics via the metricName parameter. Key Parameters:

```
labelCol: The column name containing the actual labels (ground truth).

predictionCol: The column name containing the predicted labels (output from the model).

metricName: The metric you want to use for evaluation. Common options include:
    "accuracy": Proportion of correct predictions.
    "f1": The F1 score, which balances precision and recall.
    "weightedPrecision": Precision weighted by class distribution.
    "weightedRecall": Recall weighted by class distribution.
    "weightedF1": F1 score weighted by class distribution.

accuracy: Measures the fraction of correctly predicted labels. This is useful when you have a balanced dataset.
```

Example Metrics:

```
Accuracy: Measures the overall correctness of the model.
Accuracy=Number of correct predictionsTotal number of predictions
Accuracy=Total number of predictionsNumber of correct predictions
F1 Score: A balance between precision and recall, useful when the class distribution is imbalanced.
F1=2×Precision×RecallPrecision+Recall
F1=2×Precision+RecallPrecision×Recall
Weighted Metrics: When the dataset has imbalanced classes (e.g., one class is much more frequent than the others), weighted metr
```

When to Use:

```
Model Comparison: You can use this evaluator to compare different models or hyperparameters based on their performance metrics.
Performance Monitoring: It’s useful for monitoring the performance of your model on test or validation datasets.
```

✓ 11/2/2024 (2m) 73

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
roc_auc = evaluator.evaluate(predictions)
accuracy = predictions.filter(predictions.label == predictions.prediction).count() / float(predictions.count())

print("Accuracy Score: {0:.4f}".format(accuracy))
print("ROC-AUC: {0:.4f}".format(roc_auc))
```

▸ (6) Spark Jobs

```
Accuracy Score: 0.4615
ROC-AUC: 0.2915
```

Create a Pipeline

Take transformational steps to fit into model. Basically rebuiding all the transformation steps again for the model. Using linear regression for the model.

✓ 11/2/2024 (4m) 76

```
from pyspark.ml.classification import LogisticRegression
from pyspark.ml import Pipeline

def prepare_data(df):
    df = df.select("text", "stars")
    cleaned = df.withColumn("text", lower(df.text)) \
        .withColumn("text", regexp_replace("text", "[^a-zA-Z\\s]", "")) \
        .withColumn("text", regexp_replace("text", "\\s+", " "))
    return cleaned


def create_and_train_pipeline(cleaned):

    train, test = cleaned.randomSplit([0.8, 0.1], seed=2024)
    tokenizer = Tokenizer(inputCol="text", outputCol="tokens")
    stopword_remover = StopWordsRemover(inputCol="tokens", outputCol="filtered")
    cv = CountVectorizer(vocabSize=2**16, inputCol="filtered", outputCol='cv')
    idf = IDF(inputCol='cv', outputCol="features", minDocFreq=5)
    label_encoder = StringIndexer(inputCol="stars", outputCol="label")
    lr = LogisticRegression(maxIter=100)

    # Creating the pipeline
    pipeline = Pipeline(stages=[tokenizer, stopword_remover, cv, idf, label_encoder, lr])

    # Fitting and transforming (predicting) using the pipeline
    pipeline_model = pipeline.fit(train)
    predictions = pipeline_model.transform(test)
    return predictions, pipeline_model


df = spark.read.parquet(file_location)

cleaned_data = prepare_data(df)
predictions, pipeline_model = create_and_train_pipeline(cleaned_data)

# Evaluate the model
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction", metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print(f"Test set accuracy = {accuracy}")
```

▸ (99) Spark Jobs

▾ (1) MLflow run

Logged 1 run to an experiment in MLflow. Learn more

▸ df: pyspark.sql.dataframe.DataFrame = [business_id: string, cool: long ... 7 more fields]

▸ cleaned_data: pyspark.sql.dataframe.DataFrame = [text: string, stars: double]

▸ predictions: pyspark.sql.dataframe.DataFrame

Stale widget: cannot display widget because the python repl changed. Please rerun the notebook

Stale widget: cannot display widget because the python repl changed. Please rerun the notebook

```
Test set accuracy = 0.6343853692414538
```

Save the Model file to Azure storage

Save model to storage to be applied to new yelp reviews.

11/2/2024 (10s) 79

```
# Saving model object to the /mnt/deBDProject directory. Yours name may be different.
# pipeline_model.save('abfss://de-yelp-train@storagewcdb11cc.dfs.core.windows.net/model/')
# pipeline_model.save("/mnt/deBDproject/model/")

# pipeline_model.save("/mnt/deBDproject/model/")

pipeline_model.write().overwrite().save("/mnt/deBDproject/de-yelp-train/model/")


# Save the the String Indexer to decode the encoding. We need it in the future Sentiment Analysis.
# le_model.save('abfss://de-yelp-train@storagewcdb11cc.dfs.core.windows.net/stringindexer/')
# pipeline_model.save.overwrite("/mnt/deBDproject/stringindexer/")

le_model.write().overwrite().save("/mnt/deBDproject/de-yelp-train/stringindexer/")
```

(13) Spark Jobs