Spam Email Classification using PySpark on AWS

Project Overview:

This project classifies emails as spam or ham using PySpark and AWS services. The dataset used for this analysis was acquired from Kaggle. The data processing was performed on an AWS EMR cluster, and the cleaned dataset was stored in an S3 bucket. The analysis includes text preprocessing, spam detection, and TF-IDF calculations using MapReduce.

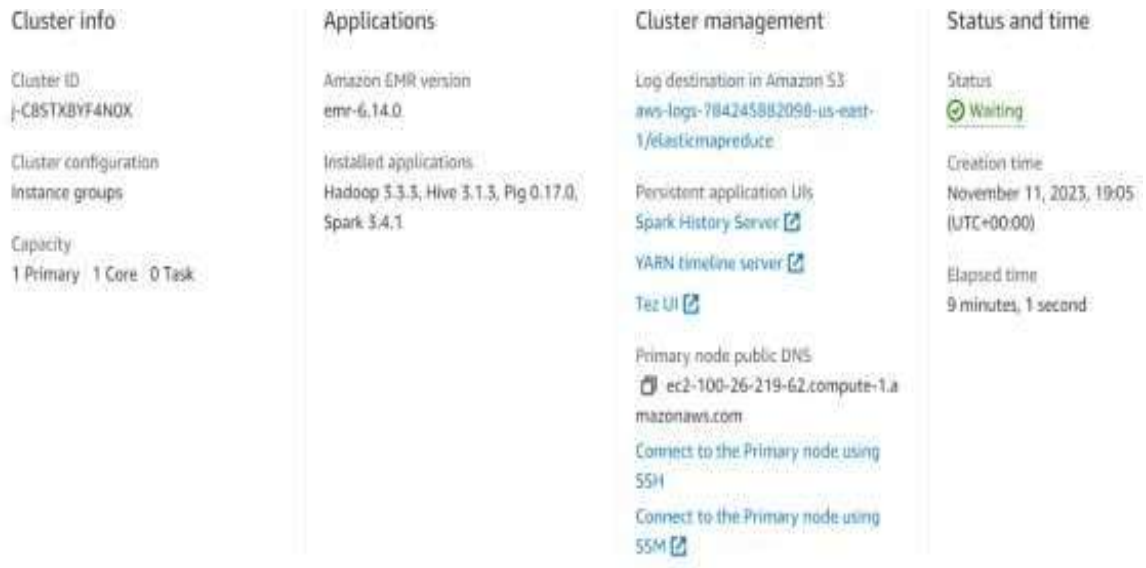
DATASET:

The dataset was obtained from Kaggle and consists of various columns such as sender email, subject, and spam indicators, which help in classifying spam and ham emails.
Dataset Link: Kaggle Spam Email Dataset

Installation & Setup:

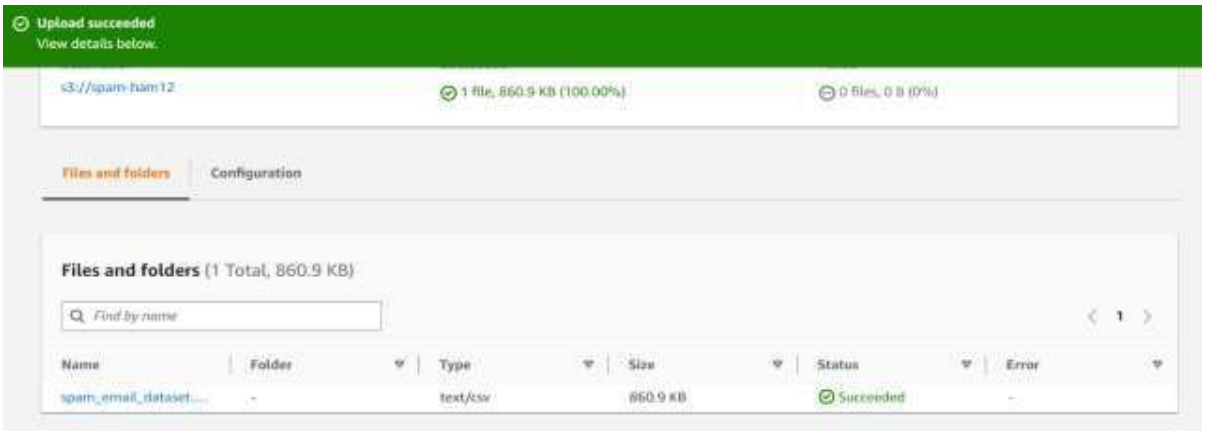
Steps to Set Up Hadoop, PIG, HIVE, and PySpark on AWS:

1. In the AWS Management Console, go to the "EMR" service and Click Create Cluster.
2. Set up a name for the multi node cluster to be created.
3. In the application bundle, select the application that needs to be installed in our case (PIG, HIVE, Hadoop, and Pyspark).
4. Configure the cluster settings, including the EC2 Instance and EC2 key pair.
5. Review the configuration settings and make sure everything is set up correctly.
6. Click "Create cluster" to launch the EMR cluster.
7. Also, add an SSH rule to the primary node in the EC2 security group inbound rules.



Steps Taken to upload the dataset in Amazon S3 Bucket:

1. The dataset is downloaded from Kaggle and uploaded to the S3 bucket by following these steps.
2. In AWS services, navigate to S3 services and create a bucket.
3. Create a bucket by choosing a globally unique name for your bucket and selecting the region.
4. After creating the bucket, choose the bucket to upload the dataset. The dataset that I have uploaded is spam_email_dataset.csv.
5. Once the upload is complete, you can see your files listed in the S3 bucket.



Steps taken for cleaning and processing the data Using pyspark:

1. I have created a cloud9 environment to run the pyspark code.
2. Pyspark is installed using the command “pyspark”.
3. A Spark session is created.

```
spark=SparkSession.builder.appName("SpamEmailAnalysis").
getOrCreate()
```

4. I have used the S3 URI link to read the csv file.
5. Below is the dataset before cleaning and preprocessing.

```
>>> df.show(5)
+--------------------+--------------------+--------------------+--------------------+----------+----------+----------+----------+----------+
|               Email|             Subject|              Sender|           Recipient|      Date|      Time|Attachments|Link Count|Word Cou
nt|Uppercase Count|Exclamation Count|Question Count|Dollar Count|Punctuation Count|HTML Tags Count|Spam Indicator|
+--------------------+--------------------+--------------------+--------------------+----------+----------+----------+----------+----------+
|mikerusso@example...|Even hotel commun...|emilyscott@exampl...|gregorysmith@exam...|13-02-2023|2023-11-11 04:13:00|         3|         0|         1
91|             32|               3|             3|           0|            11|              1|             1|
|waynebailey@examp...|Try themselves gu...|annwhite@example.net|gonzalezdaniel@ex...|09-08-2023|2023-11-11 06:15:00|         3|         9|
45|              1|               2|             1|           2|             8|              2|             1|
|  jill83@example.com|Environmental com...| david88@example.net|michellebaker@exa...|16-05-2023|2023-11-11 01:32:00|         3|         9|
52|              1|               4|             0|           2|            10|              1|             1|
|johnsonkaren@exam...|Smile real TV fat...|lindaalvarez@exam...|schroedertodd@exa...|25-04-2023|2023-11-11 14:50:00|         2|         3|
75|              6|               0|             2|           0|             8|              2|             0|
|markwilson@exampl...|Fast stage he oil...|ystafford@example...|emilywilliams@exa...|11-07-2023|2023-11-11 21:44:00|         3|         5|         2
99|            279|               0|             2|           0|             6|              3|             1|
+--------------------+--------------------+--------------------+--------------------+----------+----------+----------+----------+----------+
only showing top 5 rows
```

6. The dataset contains a total of 6,000 rows and 16 columns. There are many unwanted columns in our case. So, I have selected the columns that are required, such as sender, subject, and spam Indicator.
7. I have added a words column, where I have used Tokenizer feature which will separate and collect all the words that are used in the subject column as a list.
8. An additional column named filtered is created where the StopWordsRemover takes as input a sequence of strings (e.g., the output of a tokenizer) and drops all the stop words from the input sequences.
9. I have checked if there are any null values present in the following "Subject", "Sender", and "Spam Indicator" columns. In our case no null values are present in the dataset.
10. Below is the pre-processed data.

```
>>> df.show(5)
+--------------------+--------------------+-------------+--------------------+--------------------+
|              Sender|             Subject|Spam Indicator|               words|            filtered|
+--------------------+--------------------+-------------+--------------------+--------------------+
|emilyscott@exampl...|even hotel commun...|            1|[even, hotel, com...|[even, hotel, com...|
|annwhite@example.net|try themselves gu...|            1|[try, themselves,...|[try, guess, figh...|
| david88@example.net|environmental com...|            1|[environmental, c...|[environmental, c...|
|lindaalvarez@exam...|smile real tv fat...|            0|[smile, real, tv,...|[smile, real, tv,...|
|vstafford@example...|fast stage he oil...|            1|[fast, stage, he,...|[fast, stage, oil...|
+--------------------+--------------------+-------------+--------------------+--------------------+
only showing top 5 rows
```

Steps Taken to display top 10 Spam and Ham accounts:

1. After processing the data, the dataset is divided into two parts, spam, and ham, using the spam indicator value.
2. After dividing the dataset into spam and ham, the texts that are filtered are collected from each category respectively.

```
ham_df = df.filter(col("Spam Indicator") == 0)
spam_df = df.filter(col("Spam Indicator") == 1)
```

3. From the filtered text I have collected the top 10 most used spam and ham words that are used in the subject.

```
>>> word_counts_spam_df.orderBy(F.desc("count")).show(10)
+---------+-----+
|     word|count|
+---------+-----+
|  western|   30|
|  produce|   29|
|  control|   27|
|      buy|   25|
|investment|   24|
|     like|   24|
|   return|   24|
|      try|   24|
|     girl|   24|
|  federal|   24|
+---------+-----+
```

```
>>> word_counts_ham_df.orderBy(F.desc("count")).show(10)
+---------+-----+
|     word|count|
+---------+-----+
|    exist|   28|
|    black|   26|
| whatever|   26|
|    field|   26|
|     move|   25|
|    thing|   25|
|education|   25|
|  science|   24|
|  section|   24|
|   degree|   23|
+---------+-----+
only showing top 10 rows
```

4. After finding the top 10 ham and spam words, I have displayed the top 10 spam emails that have used the most spam words and displayed the count of how many spam words each account has used in the subject.

```
>>> top_spam_senders.show(truncate=False)
+--------------------------+-----------------------------------------------------------+-----+
|Sender                    |Subject                                                    |count|
+--------------------------+-----------------------------------------------------------+-----+
|hollandkatelyn@example.net|control control receive per.                               |2    |
|james78@example.org       |each girl yet address pick hotel.                          |2    |
|veronicaburke@example.org |address similar produce street oil.                        |2    |
|jimmy09@example.com       |board behind buy accept operation federal enjoy.           |2    |
|karinaguzman@example.org  |may up investment western since stage player.              |2    |
|katrinacross@example.org  |western strong option my investment position within.       |2    |
|hooperdakota@example.org  |federal produce peace explain you certainly everything.    |2    |
|david33@example.net       |buy job why western interesting make describe.             |2    |
|lisa57@example.org        |produce kitchen quickly mother learn return face.          |2    |
|michael20@example.com     |research theory but writer control three.                  |1    |
+--------------------------+-----------------------------------------------------------+-----+
```

5. Similarly, top 10 ham emails are displayed.

```
>>> top_ham_senders.show(truncate=False)
+--------------------------+-----------------------------------------------------------+-----+
|Sender                    |Subject                                                    |count|
+--------------------------+-----------------------------------------------------------+-----+
|ndelgado@example.com      |choose move force dream method book few.                   |2    |
|fmacdonald@example.net    |thing give move national.                                  |2    |
|meaganlarson@example.com  |us lose southern thing whatever response.                  |2    |
|jessicaallen@example.net  |exist imagine game record science skill mother.            |2    |
|mossjoshua@example.com    |enter exist help section than.                             |2    |
|laurenmcdonald@example.com|method item rule skin deep hold thing paper.               |2    |
|meganfrye@example.org     |certain step field top field see past.                     |2    |
|jasminekeller@example.org |social report education rest.                              |1    |
|susanrobinson@example.org |local field according boy could.                           |1    |
|turnerherbert@example.net |field we guy democrat customer network film.               |1    |
+--------------------------+-----------------------------------------------------------+-----+
```

Steps taken to display the TF-IDF using Map Reduce:

1. First, I have taken the top 10 spam email senders and their email subject.
2. In the next step, I split the subject text into separate words and stored them in a new column named words.
3. Then the term frequency which calculates how often each word appears in the subject.
4. All the unique words are extracted from the subject.
5. Then Inverse Document Frequency (IDF) measures the rarity of a word across all the email subjects.
6. Finally, TF-IDF is calculated by combining both TF and IDF. If a word appears often in an email but is rare across all emails, it gets a high score.
7. The TF-IDF of the top 10 spam emails is displayed.

```
>>> tfidf_df.show(10)
+--------------------+--------------------+--------------------+
|              Sender|               words|               tfidf|
+--------------------+--------------------+--------------------+
|hollandkatelyn@ex...|[control, control...|{receive -> 2.302...|
| james78@example.org|[each, girl, yet,...|{hotel. -> 2.3025...|
|veronicaburke@exa...|[address, similar...|{oil. -> 2.302585...|
| jimmy09@example.com|[board, behind, b...|{behind -> 2.3025...|
|karinaguzman@exam...|[may, up, investm...|{stage -> 2.30258...|
|katrinacross@exam...|[western, strong,...|{strong -> 2.3025...|
|hooperdakota@exam...|[federal, produce...|{explain -> 2.302...|
| david33@example.net|[buy, job, why, w...|{describe. -> 2.3...|
|  lisa57@example.org|[produce, kitchen...|{mother -> 2.3025...|
|michael20@example...|[research, theory...|{but -> 2.3025850...|
+--------------------+--------------------+--------------------+
```

8. Similarly, the same steps mentioned above are followed to show the TF-IDF of top 10 ham emails.

```
>>> tfidf_df.show(10)
+--------------------+--------------------+--------------------+
|              Sender|               words|               tfidf|
+--------------------+--------------------+--------------------+
|ndelgado@example.com|[choose, move, fo...|{move -> 1.609437...|
|fmacdonald@exampl...|[thing, give, mov...|{give -> 2.302585...|
|meaganlarson@exam...|[us, lose, southe...|{southern -> 2.30...|
|jessicaallen@exam...|[exist, imagine, ...|{exist -> 1.60943...|
|mossjoshua@exampl...|[enter, exist, he...|{exist -> 1.60943...|
|laurenmcdonald@ex...|[method, item, ru...|{deep -> 2.302585...|
|meganfrye@example...|[certain, step, f...|{see -> 2.3025850...|
|jasminekeller@exa...|[social, report, ...|{report -> 2.3025...|
|susanrobinson@exa...|[local, field, ac...|{according -> 2.3...|
|turnerherbert@exa...|[field, we, guy, ...|{democrat -> 2.30...|
+--------------------+--------------------+--------------------+
```