Big Data Project

Hadoop and Natural Language Processing Based Analysis on Kisan Call Center (KCC) Data

By:

MANKADA BATUL ABBAS

SE23MAID010

MTech – AI&DS

Project Guide:

ASSISTANT PROF. RAHUL ROY

Index

Abstract

- **Problem Statement**
  The Kisan Call Center (KCC) accumulates a vast amount of data related to farmer queries, encompassing various agricultural domains. Analyzing this data presents a significant challenge due to its volume, velocity, and variety.

- **Objective**
  This project aims to utilize Hadoop and Natural Language Processing (NLP) techniques to analyze the KCC data effectively. Specifically, the focus is on understanding the queries posed by farmers, identifying patterns, and providing insights to improve agricultural practices and farmer support services.

- **Data Description**
  The dataset used for analysis in this report is a CSV file with a size of approximately 6.7GB. It comprises several columns containing information related to queries received by the Kisan Call Center (KCC).

  The dataset is available here.

Why is it a Big Data Problem?

- ✓ **Volume**: The dataset under analysis is vast, comprising numerous farmer queries directed to the Kisan Call Center (KCC). Each query represents a distinct interaction, contributing to a substantial volume of data that necessitates thorough processing and analysis.

- ✓ **Variety**: The dataset exhibits diversity, encompassing a wide range of agricultural queries covering topics such as crop issues, pest management, fertilizer usage, and disease identification. Queries may vary significantly in language, dialect, and specific agricultural context, resulting in diverse textual data.

- ✓ **Velocity**: Data generation occurs rapidly and continuously as farmers seek assistance and information from the KCC regarding their agricultural concerns. This constant influx of queries demands real-time or near-real-time processing and analysis to provide timely insights and responses to farmers.

- ✓ **Veracity**: The dataset may contain inconsistencies, errors, or noise stemming from factors like human data entry errors, variations in farmers' language or terminology, and technical glitches during data collection. Ensuring data accuracy and reliability is crucial for obtaining meaningful insights and making informed decisions.

- ✓ **Value**: Despite challenges posed by the dataset's volume, variety, velocity, and veracity, deriving insights from the KCC dataset can yield significant value. Analysis of queries and trends can provide valuable insights into farmers' needs, challenges, and preferences, informing decision-making processes, enhancing agricultural practices, and improving support services for farmers. Ultimately, this can lead to increased agricultural productivity, sustainability, and welfare.

Big Data Tools

The paper combines MapReduce in a Hadoop environment and Natural Language Processing (NLP) clustering in PySpark for data analysis:

1. MapReduce in Hadoop:
   - Tasks:
     - Frequency of Crops Asked About
     - Frequency of Query Types
     - Frequency of Crop Categories
     - Frequency of Different Sectors
   - Approach:
     - MapReduce distributes tasks across nodes for parallel processing.
     - Mapping phase extracts relevant information from each record.
     - Reducing phase aggregates and computes frequencies.

2. NLP Clustering in PySpark:
   - Task: Grouping Similar Queries
   - Approach:
     - Data pre-processing for noise removal, case folding, and lemmatization.
     - Feature matrix creation with unique words (unigrams) and query frequencies.
     - Similarity matrix generation based on word frequency.
     - Clustering using DBSCAN for cluster estimation and agglomerative clustering for grouping queries.

Implementation

- There are 4 Map-Reduce task:
  - Implement all the 4 task using given steps bleow:

Step1:

Starting the Hadoop env by using :

```
./start-all.sh
```

Step2:

Insert all the required file i.e. data file, code file using put command:

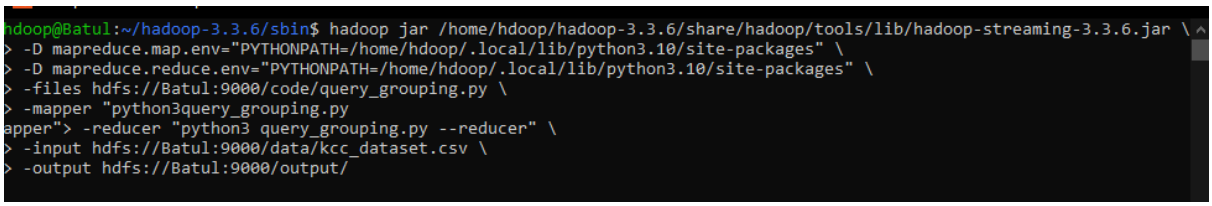
```
hdfs dfs -put -f /mnt/d/batul/Sem2/bd/project/query_grouping.py /code/query_grouping.py
```

Step3:

Executing command for python file:

As we are running python code for map reduce, in python we don't need to create jar file but we can run our python code directly.

Command:



Execution of code :

```
hdoop@Batul: ~/hadoop-3.3.6/sbin
hdoop@Batul:~/hadoop-3.3.6/sbin$ hadoop jar /home/hdoop/hadoop-3.3.6/share/hadoop/tools/lib/hadoop-streaming-3.3.6.jar -
D mapreduce.map.env="PYTHONPATH=/home/hdoop/.local/lib/python3.10/site-packages" -D mapreduce.reduce.env="PYTHONPATH=/ho
me/hdoop/.local/lib/python3.10/site-packages" -files hdfs://Batul:9000/code/queryType_freq.py -mapper "python3 queryType
_freq.py --mapper" -reducer "python3 queryType_freq.py --reducer" -input hdfs://Batul:9000/data/partaa -output hdfs://Ba
tul:9000/output1/
2024-05-07 21:46:10,508 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using buil
tin-java classes where applicable
packageJobJar: [/tmp/hadoop-unjar13708741715073890757/] [] /tmp/streamjob6436388694370578449.jar tmpDir=null
2024-05-07 21:46:11,650 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-05-07 21:46:11,881 INFO client.DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2024-05-07 21:46:12,221 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/
hdoop/.staging/job_1715097735109_0002
2024-05-07 21:46:13,531 INFO mapred.FileInputFormat: Total input files to process : 1
2024-05-07 21:46:14,688 INFO mapreduce.JobSubmitter: number of splits:2
2024-05-07 21:46:15,739 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1715097735109_0002
2024-05-07 21:46:15,740 INFO mapreduce.JobSubmitter: Executing with tokens: []
2024-05-07 21:46:16,168 INFO conf.Configuration: resource-types.xml not found
2024-05-07 21:46:16,169 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2024-05-07 21:46:16,244 INFO impl.YarnClientImpl: Submitted application application_1715097735109_0002
2024-05-07 21:46:16,283 INFO mapreduce.Job: The url to track the job: http://Batul.:8088/proxy/application_1715097735109
_0002/
2024-05-07 21:46:16,285 INFO mapreduce.Job: Running job: job_1715097735109_0002
2024-05-07 21:46:24,527 INFO mapreduce.Job: Job job_1715097735109_0002 running in uber mode : false
2024-05-07 21:46:24,531 INFO mapreduce.Job:  map 0% reduce 0%
2024-05-07 21:46:36,756 INFO mapreduce.Job:  map 100% reduce 0%
2024-05-07 21:46:44,868 INFO mapreduce.Job:  map 100% reduce 100%
2024-05-07 21:46:48,001 INFO mapreduce.Job: Job job_1715097735109_0002 completed successfully
2024-05-07 21:46:48,191 INFO mapreduce.Job: Counters: 55
```

Step4:

See the output by command:

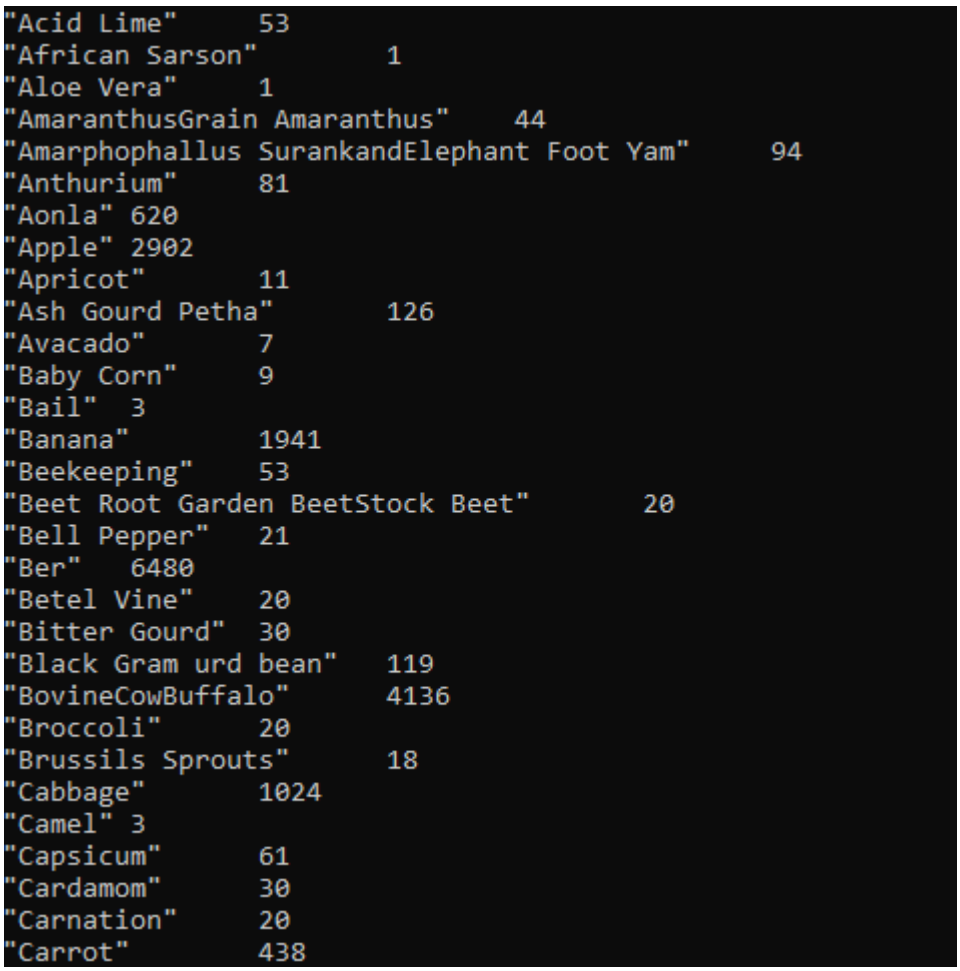
```
hdfs dfs -cat /output1/part-00000
```

Output of Query Type frequency count:

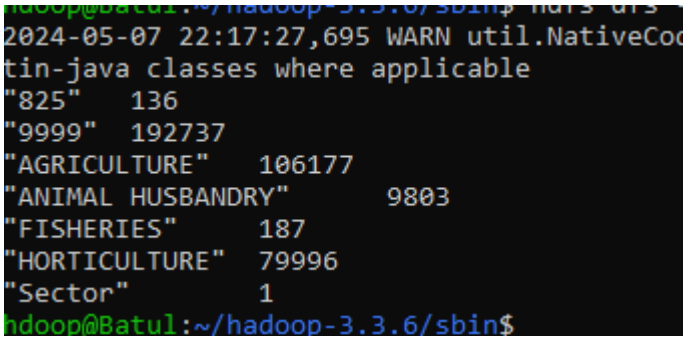
```
'Agriculture Mechanization"     1250
'Credit"        76
'Cultivation Conditions"        18
'Cultural Practices"    419
'Dairy Production"      632
'Disease Management"    1813
'Disease"       2
'Feed"  2
'Fertilizer Use and Availability"       14223
'Field Preparation"     24
'Loans "        3
'Management"    3
'Market Information"    315
'Others"        45
'Plant Protection"      75
'Poultry"       31
'QueryType"     1
'Soil Testing"  260
'Training"      4
'Varities"      4
'Water Management Micro Irrigation"     31
'Water Management"      420
'Weather"       3608
```

Same way,

Output of Crop Frequency Count:



Output of Sector Query Frequency Count:



- NLP Query Clustering PySpark Task :
    - PySpark is the Python API for Apache Spark, a fast and general-purpose cluster computing system. It provides high-level APIs in Python, Java, Scala, and R, making it easier to build parallel applications to process large-scale data sets.

Importance of PySpark for the Task:

- Distributed Computing: PySpark enables the processing of large-scale datasets by distributing computations across a cluster of machines.
- Parallelism: It leverages parallel processing to perform operations in-memory, leading to faster processing of data.
- Scalability: PySpark can scale horizontally by adding more nodes to the cluster, allowing it to handle growing volumes of data.
- Ease of Use: PySpark provides high-level APIs and libraries for data processing, machine learning, and streaming analytics, making it accessible to data scientists and engineers.
- Integration: It seamlessly integrates with other big data technologies like Hadoop, HDFS, Hive, and HBase, enabling interoperability with existing data infrastructures.

Summary of the Code:

I use PySpark for preprocessing textual data, performing NLP-based clustering, and visualizing the results. It loads a dataset of farmer queries, preprocesses the text using tokenization and lemmatization, transforms it into TF-IDF vectors, and clusters similar queries using the KMeans algorithm. Finally, it visualizes the clustering results to gain insights into the queries received by the Kisan Call Center. PySpark's distributed computing capabilities make it well-suited for processing large volumes of textual data and performing complex analytics tasks efficiently.

Resuls of all the above task done in pyspark:

```
+-----------------+-----+
|        QueryType|count|
+-----------------+-----+
|               51|   17|
|               15|  185|
|               11|  114|
|          Poultry|   31|
|               29|41086|
|               87|  140|
| Plant Protection|   70|
|                3|22315|
|Field Preparation|   23|
|               34|   14|
+-----------------+-----+
only showing top 10 rows
```

```
+------------+------+
|        Crop| count|
+------------+------+
|        9999|166704|
|Cotton Kapas| 19503|
|       Wheat| 14110|
|      Tomato|  7550|
|      Others|  7211|
|        1280|  7192|
|        1279|  6419|
|       Onion|  6023|
|         Ber|  5646|
|        1037|  5040|
+------------+------+
only showing top 10 rows
```

```
+-----------+------+
|   Category| count|
+-----------+------+
|          0|332261|
|Fiber Crops|     1|
| Vegetables|     1|
+-----------+------+
```

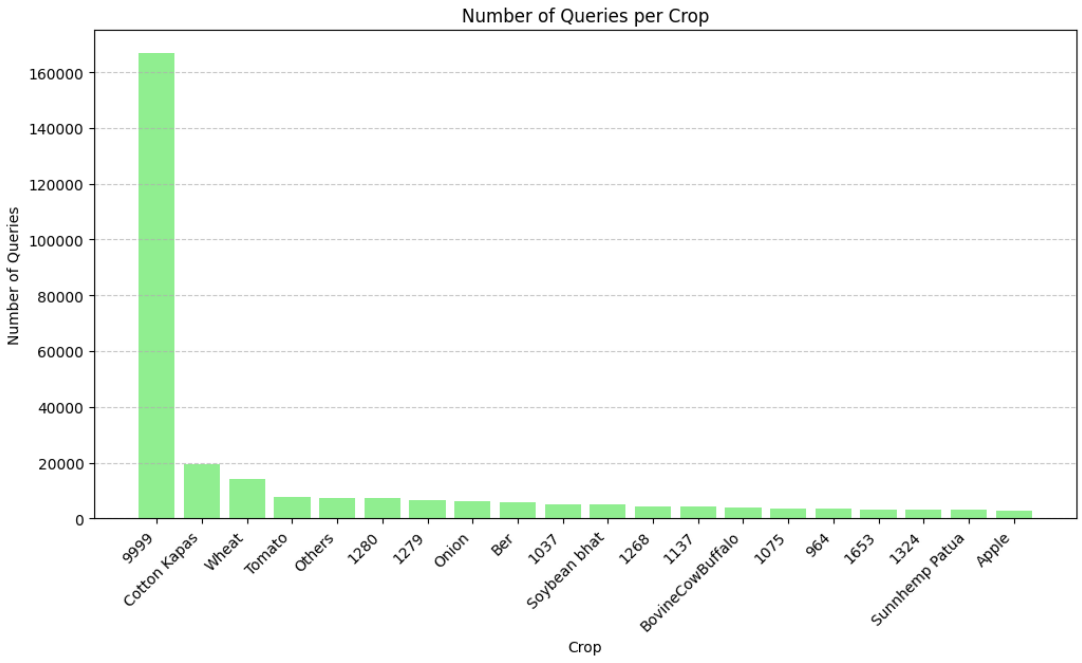
```
+----------------+------+
|          Sector| count|
+----------------+------+
|ANIMAL HUSBANDRY|  9000|
|    HORTICULTURE| 69911|
|             825|   134|
|       FISHERIES|   185|
|     AGRICULTURE| 86329|
|            9999|166704|
+----------------+------+
```

Results of Clustered Queries:

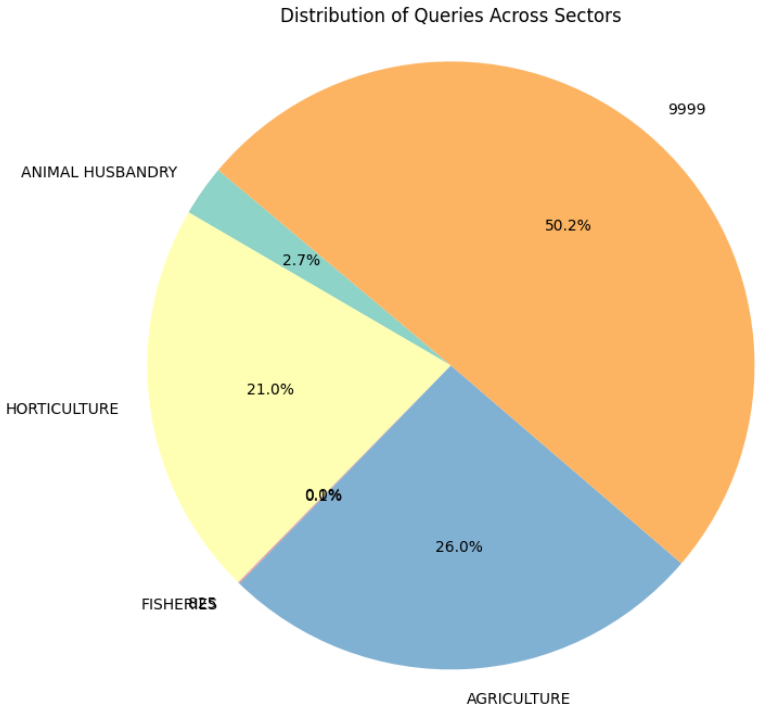
QueryText	prediction
how to control flower drop in bottelgourd	0
how tyo control diseases in buffalo	0
how to control fruit borer in brinjal	3
how to control of yellow moisac in moong	0
how to control white fly in brinjal	0
how to control termite in wheat	0
how to control chilli thrips	0
how to increase milk in buffalo	0
how to control sucking pest in bottle gourd	1
what is the control of the semilooper in the soybean	0
how to control becterial blight in tomato	4
what is the control of the pest in sugarcane	0
what is the control of the Bihar hairy caterpillar	0
what are control of cow pea mosaic	0
what is the control of the Tobacco caterpillar	0
what is the control of the insect in coriander	0
what is the important varieties of the cabbage for the madhya pradesh	0
what is the control of the semilooper in the soybean	0
how to control yellow moisac in moong	0
how to control white fly in brinjal	0

only showing top 20 rows

No. of Queries vs Crop:



Pie Chart od queries across sector:



Conclusion:

- In conclusion, the analysis of Kisan Call Center (KCC) data using big data and natural language processing (NLP) techniques has provided valuable insights into the queries raised by farmers.

- Through the application of MapReduce in Hadoop environment, we were able to perform various tasks such as determining the frequency of different crops, query types, crop categories, and sectors. This allowed us to understand the prevalent concerns and topics of interest among farmers.

- Additionally, by leveraging PySpark, we conducted NLP-based clustering to group similar queries. This approach helped identify common themes and questions asked by farmers, enabling more efficient handling of queries and provision of relevant information through KCC.

- Overall, the combination of big data analytics and NLP techniques offers significant potential for enhancing the effectiveness and responsiveness of agricultural support systems like KCC. By gaining deeper insights into farmer queries and concerns, agricultural authorities can better tailor their services and interventions to meet the needs of farmers, ultimately contributing to improved agricultural productivity and livelihoods.