Selection of Favorite Recipes using Energy Classification by MapReduce

Sankara Subramanian Venkatraman
School of Computing
National College of Ireland
Dublin, Ireland
x18179541@student.ncirl.ie

***Abstract*—This project presents a MapReduce algorithm for analyzing the favorite recipes of countries under the European Union (EU28) based on their energy product classification. Factors like geo-location, standard international energy product classification (SIEC), ingredients and number of ingredients for preparing the recipe are used in the analysis.**

OpenStack, a private cloud provided by National College of Ireland, is provisioned for performing the MapReduce algorithm in Java. Python is used for performing ETL and cleaning the datasets. Input and Output datasets are stored, retrieved and processed using Hadoop Distributed File System (HDFS). Pearson correlation (r) is calculated to identify correlation between energy value and number of ingredients. Finally, the favorite recipe of countries for each year based on energy product classification is discovered.

***Index Terms*—MapReduce, Java, Big Data, Hadoop, OpenStack.**

I. Introduction

Significant growth in the volume of digital data emphasizes the importance of big data [1]. MapReduce is a programming model developed by Google for processing large intensive data while maintaining high performance, parallel computation and fault tolerance [2]. Python is an open-source platform for implementing MapReduce. Python can also be executed in high-performance computing (HPC) environments [3].

In this project, Python cleans the datasets using Pandas and Numpy libraries. It transforms raw data into conventional data for analysis.

The main objective of this project is to identify the favorite recipe of EU countries based on energy product classification and the country with maximum recipes for each year.

Correlation between energy dataset and recipe dataset is evaluated using the Pearson correlation estimator. The correlation coefficient (r) is calculated based on the energy value and the number of ingredients. This will check if the increase in energy value, will affect the number of ingredients of the recipes.

Data cleaning, transformation, scaling issues faced during execution, and the future scope of this project are discussed in the later sections.

Fig. 1. Entity Attribute Diagram

II. Data Sources

In this section, data sources and metadata of energy product classification and recipes are presented. The datasets are sourced from eurostat [7] and food [8] websites.

A. Energy Product Classification dataset

The first dataset has information about supply, transformation, and consumption of oil and petroleum products for EU countries. It contains 40 types of energy product classification. The dataset contains **58800** records. The metadata contains

1) Energy balance
2) Standard international energy product classification (SIEC)
3) Unit of measurement of energy consumption (in THS_T)
4) Geographical location

B. Recipe dataset

The second dataset has information about names, ingredients, cooking time and description of the recipes. The dataset contains **231636** records. The metadata contains

1) Recipes name
2) Time taken (in Minutes)
3) List of Steps
4) Number of steps to cook
5) Description
6) Ingredients
7) Number of ingredients

From the above two datasets, the following insights are discovered and previous research works of the datasets are discussed;

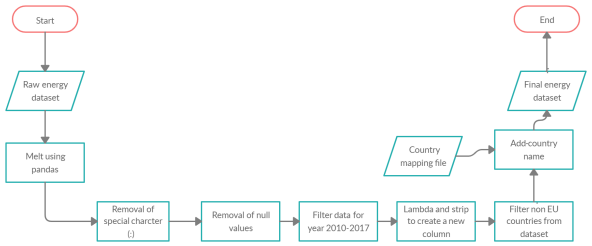

Fig. 2. Flow Chart for Energy Dataset

i. With these datasets a spurious correlation between energy product classification code and the favorite recipe of a country with year as the common attribute is determined.
ii. Even though there is no relationship between the datasets, an attempt is made to discover a hidden pattern.
iii. Fig.1. represents the Entity-Attribute diagram for recipes and energy datasets. The attributes used for the analysis are included in the figure.
iv. The primary key attribute **(key_column)** in both datasets is underlined.
v. The key_column in **Energy dataset** is formed by concatenating year and energy code.
vi. The key_column in **Recipe dataset** is formed by concatenating year and number of ingredients.

C. Previous Related Works

A study generated personalized recipe from historical user preferences. The strategies such as instructional texts, quality and quantitative measures are leveraged. Strategies such as prior technique attention and attention fusion layer are incorporated to generate the favorite recipe on user preference and input specification [4].

Another study developed a model to set the targets of energy policy for EU countries based on energy consumption and economic growth. The relationship between energy and economic growth are analyzed using various hypotheses such as the growth, conservative, feedback and neutrality hypotheses. The countries are divided into 4 groups and the bootstrap model is applied to find the causality between energy consumption and economic growth was observed. Finally, it proposed that energy policy reforms should not affect the economy [11].

III. Methodology

The project follows proper data quality control measures of big data; Define, Assess, Analyze and Improve [5]. The dimensions of data quality explained in the following sections;

A. Intrinsic:

Accuracy of the datasets is obtained by removing the blank records, duplicate ids, and outliers and is shown in Fig.3., Fig.4, Fig.5. and Fig.6. The energy dataset has an intense volume of records and the recipe dataset has complex data types like an array.

```
In [15]: energyDataFinal.head(10)
```

Out[15]:

	nrg_bal	siec	unit	year	energy_val	siec_sum	Country_Name
0	BKFLOW	O4000	THS_T	2017	0.045	4	Netherlands
1	BKFLOW	O4000XBIO	THS_T	2017	0.045	4	Netherlands
2	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	1402.700	20	Belgium
3	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	78.000	20	Czechia
4	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	6030.000	20	Germany
5	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	75.000	20	Spain
6	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	203.000	20	Finland
7	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	1215.000	20	France
8	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	236.000	20	Hungary
9	BKFLOW	O4100_TOT_4200-4500	THS_T	2017	1776.000	20	Italy

Fig. 3. Final Energy Dataset

```
In [5]: rawRecipes['submitted']= pd.to_datetime(rawRecipes['submitted'])
        rawRecipes['year'] = pd.DatetimeIndex(rawRecipes['submitted']).year
        rawRecipes=rawRecipes[rawRecipes['year'].between(2010,2017)]

In [6]: rawRecipes.head(2)
```

Out[6]:

	name	id	minutes	contributor_id	submitted	tags	nutrition	n_steps	steps	description	ingredients	n_ingredients	year
115	1 in canada chocolate chip cookies	453467	45	1848091	2011-04-11	['60-minutes-or-less', 'time-to-make', 'cuisin...	[595.1, 46.0, 211.0, 22.0, 13.0, 51.0, 26.0]	12	['pre-heat oven the 350 degrees f', 'in a mixi...	this is the recipe that we use at my school ca...	['white sugar', 'brown sugar', 'salt', 'margar...	11	2011
125	2000 meatloaf	475785	90	2202916	2012-03-06	['time-to-make', 'course', 'main-ingredient', ...	[267.0, 30.0, 12.0, 12.0, 29.0, 48.0, 2.0]	17	['pan fry bacon , and set aside on a paper tow...	ready, set, cook! special edition contest entr...	['meatloaf mixture', 'unsmoked bacon', 'goat c...	13	2012

Fig. 4. Final Recipe Dataset

Robustness of the dataset is defined by having a wide range of values for energy value, energy code, year and number of ingredients.

Reputation is based on third-party experience [6]. The datasets are sourced from a trusted website and are reliable. Datasets sourced from Eurostat and Food.com.

B. Contextual:

Attributes such as key_column, energy_val, siec, country_name are present in energy dataset. In the recipe dataset, n_ingredients and recipe_name provide value to the analysis and are beneficial.

The attributes in the dataset contain no null values and are complete. Data, variables, number of records and concepts of the dataset are explained in the data source section. The data source section explains the relevancy of the dataset.

The volume of the data is quite huge with 58800 in the energy dataset and 231636 records in the recipe dataset.

C. Representational:

In both the datasets, the data is an appropriate, unit of data, and the attributes are defined clearly. The ambiguity of the dataset is cleared during the cleaning process to avoid complexity.

```
In [488]: sns.boxplot(x=energyData_o['energy_val'])
Out[488]: <matplotlib.axes._subplots.AxesSubplot at 0x1d107575288>
```

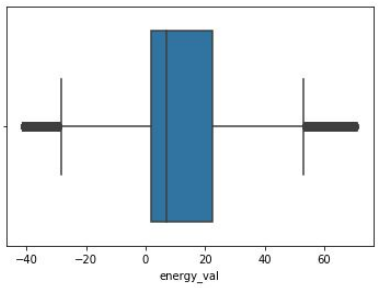

Fig. 5. Energy Dataset Without Outliers

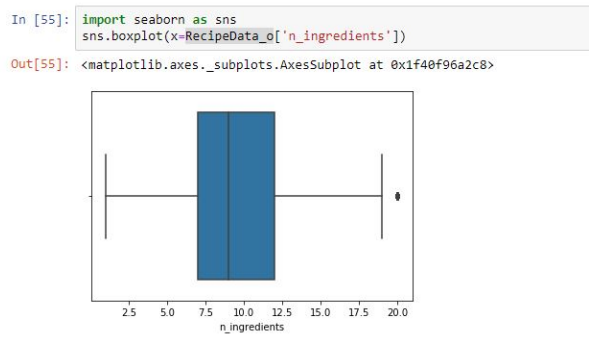

Fig. 6. Recipe Dataset Without Outliers

D. Accessibility:

Datasets are stored in HDFS in the directory /**user**/**project**/ and are easily accessible through Hadoop commands. The security of the datasets is ensured by storing in Openstack instance. A valid .pem or .ppk file can authorize the machine and utilize the dataset.

IV. Implementation and Architecture

1. Architecture

- This project uses **OpenStack**, a private cloud instance, for the computation of intensive datasets.
- Based on the complexity of the datasets, instance type is selected.

  Machine Specification:

  - Machine_Type - Ubuntu-Bionic 18.04.3 LTS
  - RAM - 4 GB
  - Virtual CPUs - 2
  - Hard disk - 40 GB
  - Availability Zone - nova
- Floating IP associated with the Openstack instance x18179541_DIA_Project is 87.44.4.162.
- Java 8, OpenJDK version "1.8.0_222" is installed in the machine for configuring Hadoop.
- Hadoop - 3.1.3 is installed in the Openstack instance for performing MapReduce.
- User **hduser** is created to install and set up Hadoop environment with in the Openstack instance.

2. Implementation

Python has many in-built library packages for cleaning datasets such as numpy, pandas, map, lambda, scipy, seaborn, matplotlib, and regex, etc.

A. Energy dataset

Fig.2. represents the flow chart of the energy dataset. Using Jupyter Notebook, Pandas is imported, and a tab-separated file is read using pd.read_csv function.

Melt function in Pandas is used to convert year and energy_val attributes from columns to rows, based on nrg_bal, siec, unit and country_code as id_vars. Using str.contains function of python, special character (:) is removed by generating a "flag" column. energy_val = 0 is also removed from the dataset.

Numpy is used to extract unique value for year attribute and flag column created in the previous step is dropped. The attribute Year is converted to an integer and for analysis, we take year from 2010 to 2017.

In the next step segregating numeric value from the SIEC column of energy dataset using lambda and map by stripping (_) and (-) and all the alphabets. Add all the digits of segregated numeric value and store it in new column siec_sum.

Concatenate year and siec_sum column to obtain key_column using Microsoft Excel. Finally, a dataset with key_column, siec, unit, energy_val, country_name is created.

The outliers are removed and normalization of z-score >3 for energy value is removed and shown in Fig.5.

B. Recipe dataset

Fig.7. represents the flow chart of the recipe dataset. Comma-separated file is read using pd.read_csv function.

Extract year (yyyy) from "submitted" attribute which contains data in (MM/DD/yyyy) using DatetimeIndex from Pandas and year from 2010 to 2017 is taken for analysis.

Concatenate year column with number of ingredients (n_ingredients) to obtain key_column using Microsoft Excel. Finally, a dataset with key_column and recipe_name is created.

C. Map Reduce Join Dataset

MapReduce Algorithm is widely used to process intensive datasets by utilizing a key-value pair framework and it computes the data in distributed and parallel computing. MapReduce is highly scalable and executes the program in 3 stages (i.e) map, shuffle and reduce.

EnergyMapper.java and RecipeMapper.java are the two mapper files which extract data from HDFS and segregate data as (key-value) pair. In both mapper files **key_column** is taken as key and the remaining columns are considered as value.

ReduceJoinReducer.java performs the reducer phase in which both the datasets are joined based on the key_column. Tag and data separator are used to split the values and process the data. Two private array list is created to store the dataset of recipe and energy. If both the array is not null, then join the list and create a new joined dataset.

ReduceJoin.jar is the configuration file to run the map-reduce process which takes input from HDFS. The ideas were inspired by [9].

D. Pearson Correlation Dataset

Pearson correlation (r) is used to determine the relationship between the energy and the recipe dataset. The columns used to find the correlation are **energy_val** from energy dataset and **n_ingredients** from recipe dataset for every value of key_column.

In the final output of Pearson's job, we found 7 values. The sum of energy_value for the joined dataset is 2.6846E8. The sum of n_ingredients of the joined dataset is 1.5862E8.

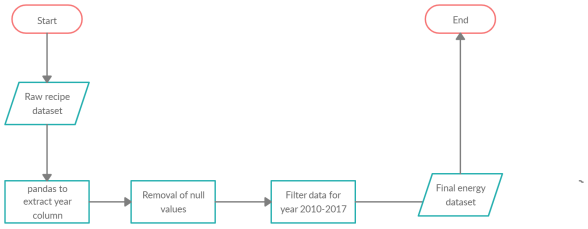

Fig. 7. Flow Chart for Recipe Dataset

The squares of energy_value are 9.9431E9 and squares of n_ingredients are 1.7826E9. The product of energy_value and n_ingredients is 2.5029E9. The total number of records in the dataset is 1.6600E7.

Using the above 6 values, the Pearson correlation (r) of the dataset is determined. The value is **-0.0728**. The correlation is negative and minimum [11].

E. Grouped Map Reduce Dataset

Fig.8. represents the Entity Relation (ER diagram) of the final grouped map-reduce data model from the recipe and energy dataset.

The favorite recipe and count of recipes of each country based on key_column is obtained from GroupMR.java file.

It contains one mapper, one reducer and the main class along with custom sorting class on the country and key_column. The idea of using group-by based on a composite key is inspired from [10].

V. Results

A. Architecture

- Installation of Hadoop in Ubuntu 18.04 as a hduser user is a challenging task.
- Starting master & slave nodes and resource manager will enable Hadoop services. But, while shutting down the instance and restarting the services namenode is not started.
- To fix this issue, hadoop namenode -format is used, which is not a permanent solution.
- Also, Yarn-a resource manager service- is not displayed, after starting the service.
- The above two issues are because Hadoop 3.1.3 version is not compatible with Java 11. So this issue is fixed by replacing Java 11 with **Java 8**.

B. Evaluation and Dataset Results

- The initial raw tab-delimited energy dataset contains 58,830 records. After cleaning and transformation, the dataset grows to 133895.
- The initial comma-separated recipe dataset contains 231636 records. After cleaning and transformation, the dataset is reduced to 30009.
- The output of the joined dataset contains 3.6 million records is cross-verified using Pandas merge function to check whether the dataset has performed a Cartesian (cross-join) join. But it returns the same result as MapReduce. Fig.5. represents the final count after joining the datasets.

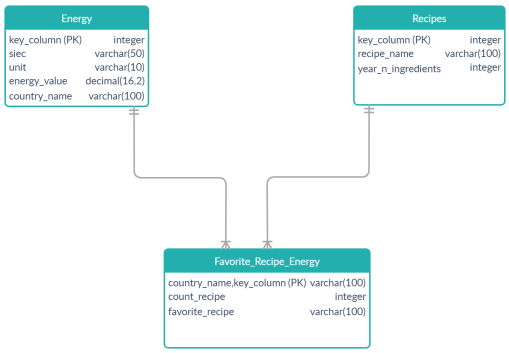

Fig. 8. ER Diagram

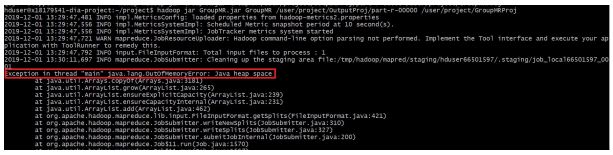

Fig. 9. Error Message

- While trying to execute GroupMR.jar with 3.5 million from joined dataset, it throws an error of Java heap space, which is out of Memory error. Fig.9. shows the error message.
- The actual size of the joined dataset is 2.2 gigabytes.
- In the first step, the first 1000 records from the joined dataset is extracted and code is run. It is executed and output is generated within a minute.
- In the second step, the first 5000 records are tested with the same code, the output is generated in 20 minutes and record count of the output is 1226.
- We cannot conclude that for the year 2010 Netherlands' favorite recipe is **avocado mascarpone dream cream dessert**.
- In the third step, random 5000 records is chosen from the dataset using **shuf** command in Linux.

C. Insights Discovery

- The dataset with a randomly chosen 7000 records is executed in 33 minutes and the record count of the output is 1446.
- 10000 records were randomly chosen from a joined dataset and executed. It ran for 2 hours 58 minutes and again faced the error of java heap space.
- The interesting part of this project is to find the favorite of a country for a particular year based on energy classification code is achieved.
- But the key challenge of is to find the favorite recipe for all countries and for all energy code from 2010 to 2017 is not achieved because of scaling issues.

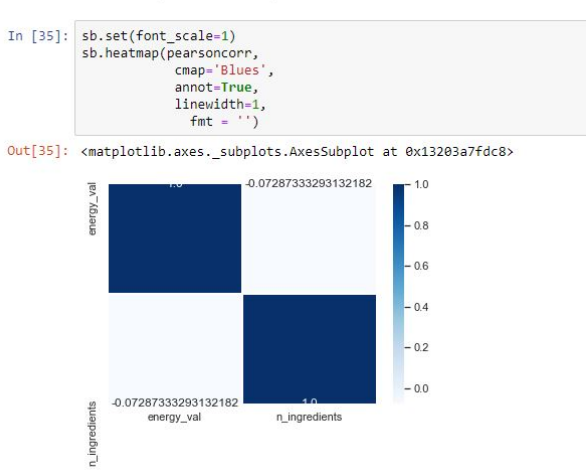

Fig. 10. Pearson correlation Plot

- After replacing the **java heap memory to 2 GB** for random 10000 records it took 1 min 20 seconds to complete the task.
- Pearson correlation (r) value is negative and weak, so the **energy value is not a good predictor of increase in the number of ingredients of a recipe**.
- Pearson correlation plot for energy value and the number of ingredients is shown in Fig.10. The negative correlation and the value says there is no relationship between the attributes.
- From 3.6 million, 0.1 million datasets are drawn randomly. The function GroupMR is run to find the favorite recipe. The code took 22 minutes 19 seconds to complete the task with 2 GB heap space.
  The benchmark of the dataset for different configuration is represented in Fig.11.

VI. Conclusion and Future Work

A. Conclusion

In this project, various aspects of data-intensive components are utilized such as Java as the programming language, MapReduce model for executing code chunks and Hadoop Distributed File System for Storing the datasets. Being a single node cluster, the program is not executed as expected for 3.6 million records.

The next step is to increase the number of instances to 2 or use a machine with a higher configuration than **m1.medium**. We can use the m1.large machine which has 4 CPUs, 8 GB RAM and 80 GB hard disk for better performance.

Also, restricting the raw dataset size to 10000 and try executing with the same configuration for the year 2015 to 2017 may provide results with better accuracy. Pearson correlation (r) can be improved by limiting the energy value range from 100 to 200.

Fig. 11. Benchmark configuration

B. Future Work

Text analytics can be performed in the dataset to find the favorite recipe of a country. If suppose, the country's first 3 letters or last 3 letters are available in the ingredients columns of the recipe dataset then these can determined as the favorite recipes of the country.

Map Reduce algorithm can be performed using Pig to load the data into HDFS. Hive Query Language (HQL) will execute the join operation and groupby functionalities which is similar to SQL. Finally, we can store the data in HDFS or Hbase which is a row-columnar database.

Spark by default process data in memory. It can also be used for processing the dataset and perform joins and groupby with the help of dataframe and finally the data can be stored in HDFS.

References

[1] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC iView: IDC Analyze Future, vol. 2007, pp. 1–16, 2012.
[2] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.
[3] T. D. Hartley, U. Catalyurek, A. Ruiz, F. Igual, R. Mayo, and M. Ujaldon, "Biomedical image analysis on a cooperative cluster of gpus and multicores," in ACM International Conference on Supercomputing 25th Anniversary Volume. ACM, 2014, pp. 413–423.
[4] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating Personalized Recipes from Historical User Preferences," 2019.
[5] K. Wang, Cloud Computing for Machine Learning and Cognitive Applications: A Machine Learning Approach. 2017.
[6] F. Leal, "Seminar on Trust & Reputation using Crowdsourcing Data."
[7] https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=nrg_cb_oil&lang=en
[8] https://Food.com Recipes and Interactions
[9] https://javadeveloperzone.com/hadoop/reduce-side-join-mapreduce-example-using-java/
[10] https://techmytalk.com/2014/11/14/mapreduce-composite-key-operation-part2/
[11] https://vangjee.wordpress.com/2012/02/29/computing-pearson-correlation-using-hadoops-mapreduce-mr-paradigm/
[12] S. Śmiech and M. Papiez, "Energy consumption and economic growth in the light of meeting the targets of energy policy in the EU," Int. Conf. Eur. Energy Mark. EEM, pp. 1–5, 2014.