Assignment 2: App Construction

CA675: Cloud Technologies

Final Report

Group 12

19210249	Himanshu Vashisht (GC)	himanshu.vashisht2@mail.dcu.ie
19210509	Nikhil Mittal	nikhil.mittal2@mail.dcu.ie
19211007	Yashaswi Verma	yashaswi.verma2@mail.dcu.ie
19210256	Alekhya Nethra	alekhya.nethra2@mail.dcu.ie
19210189	Bhargav Dandu	bhargava.dandu2@mail.dcu.ie
19210222	Joseph Cherian	joseph.cherian2@mail.dcu.ie
19210135	Akanksha Rajpute	akanksha.rajpute4@mail.dcu.ie

School of Computing, Dublin City University

Introduction

The project is a wine analytics dashboard that allows the users to see different varieties of wines prevalent across 7 different countries based on features like province, variety, price and the points. The wine dataset was particularly taken into consideration for the simple fact that it can give a theoretical analysis as to which wine is better of the lot across 7 major wine producing countries of the world. The application made use of a dataset publicly available from Kaggle[1] and it also included several features such as variety, price, location, points, etc. which played a key role in the choice of wine by different individuals across this country. Hence, it laid a strong foundation in the meaningful analysis.

We developed a web application where users can see the appropriate analysis, we have done by the means of knowing wine varieties across these 7 countries. Also, the web app informs the user about the wines that are famous in that country. Further appropriate visualizations are shown based on the wine variety, price, location, ratings, etc. using visualisations such as geographical maps.

URL for video:

https://youtu.be/KR-VlB9YZWI

Github Repository:

https://github.com/him89088/CA675_Assignment2

Technologies used

The main technologies that were used are as follows:

Task	Technology
Data Cleaning	Python
Data Loading/Querying	Apache Spark/Scala
Visualising	Tableau
Frontend	HTML, CSS, Bootstrap, JS, jQuery
Backend	Python Flask, Tableau-Js-API
Deployment	Docker, Kubernetes

Motivation behind Technologies Used:

Data Cleaning

Apache Spark(Scala)

- It is 100 times faster than hadoop
- Apache Spark carries easy-to-use APIs for operating on large datasets
- Apache Spark supports many languages for code writing such as Python, Java, Scala, etc.

Python

- We wanted to Tokenize the data
- It has multiple methods for data filtering.

Cloud Storage

Bucket[2]

- There is a single namespace shared by all buckets
- Buckets contain objects which can be accessed by their own methods.

Data Querying

Google BigQuery[3]

- Its Flexible Architecture Speeds Up Queries
- Access the Data You Need on Demand

Data Visualizations

Tableau[4]

- Remarkable Visualization Capabilities
- Ease of Use & Implementation
- Handles large amounts of data.

Front End
HTML

- It is Supported by all Browsers.
- It can Integrate Easily with Other Languages.
- It is Lightweight.

CSS & Bootstrap

- Responsible for web app designing

JavaScript

- Extended functionality to web pages.
- No compilation needed.
- Platform independent.

Backend
Flask[5]

- Integrated support for unit testing
- Built-in development server and fast debugger
- Restful request dispatching.

Containers
Docker[6]

- Because of its size and minimal requirements rapid application deployment can be done.
- Portability across machines can be done using a single container.
- It helps in sharing containers using a remote repository

Application Deployment
Kubernetes[7]

- Kubernetes is declarative as it describes the state of cluster.
- Kubernetes can run on virtually any public cloud, on-premise hardware, or even bare metal.
- Kubernetes determines which Worker Nodes a container should run on based on available resources which optimizes resource usage.

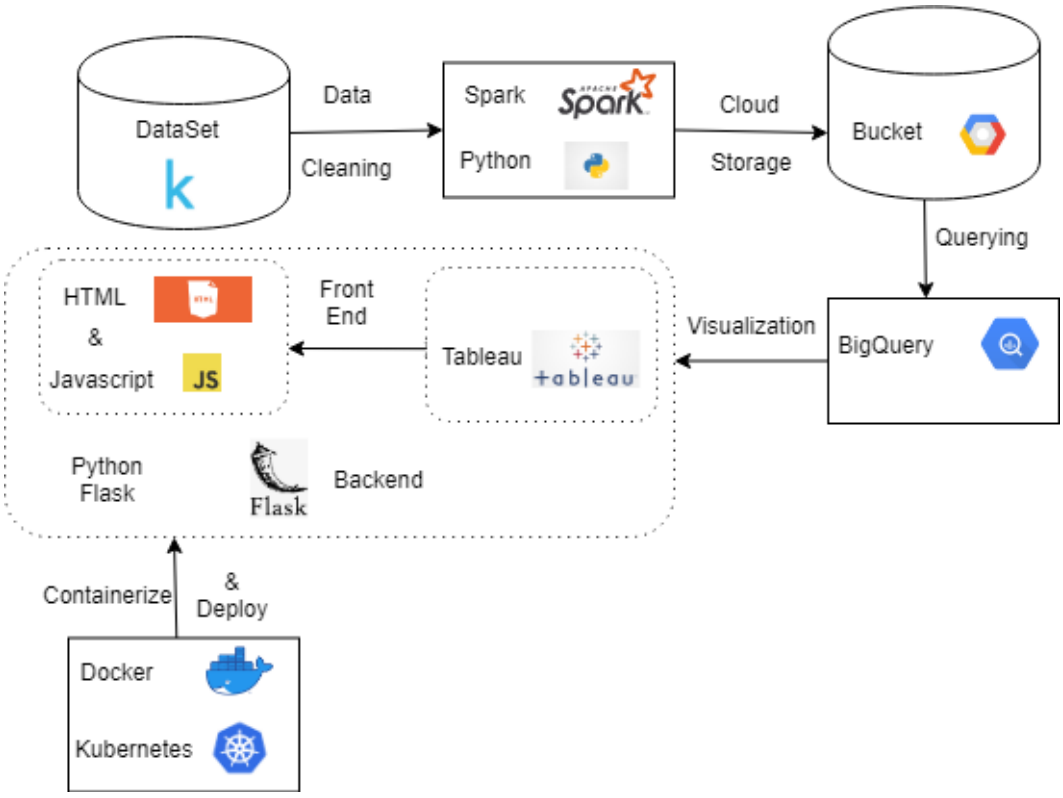

Figure 1: Data Flow Diagram

Data: Source Selection, Preparation & Cleaning

Having an interesting dataset to work on is the first key step in creating captivating analysis. A key emphasis was laid on finding the right dataset that would borrow implementations of both the Big data theories and various cloud processing technologies. We took the wine dataset for its variety of features and aim to provide a theoretical analysis by the means of apt visualisations.

The dataset which we are using in our Web App is taken from Kaggle. The dataset represents the wine reviews data collected from different countries that have different attributes for the ratings of wine and much more information. The data set is a CSV file of 50.6 Mb, having 16 columns and approximately 130 thousand rows. The columns contain information such as Country, Points, Price, etc.

Firstly ,the dataset which we acquired was loaded in Hadoop Cluster so that the file is in the HDFS format. Then we loaded the data into spark from HDFS, so that we can process and clean data. The dataset had Null values in different columns, which we cleaned using spark-shell. Some columns were of no use in the visualization phase of the project, so we scoured the data accordingly and stored it in SPARK.

Data Processing, Querying & Storing

The cleaned data was then stored into a bucket on Google Cloud Platform., which was streamlined to Google BigQuery. Google BigQuery was connected to Tableau and all the analytics was done on Tableau.

Data Visualization Trends

The wine dataset was explored and visualised through comparative visualizations such as geographical maps, bubble plots etc. to deduce the wine variety prevalent in a country. Also, the pricing and popularity through its ratings which was shown by points was taken into account while showing trends. For this, we used Tableau to create comparative data visualizations and graphs for better analysis.

Firstly, appropriate colour coding was given for each of the 7 countries. Furthermore, three appropriate visualisation graphs were drawn out of the dataset. The first graph showed a geographical world map displaying the average points and pricing for different provinces within a country. ‘Symbol Maps’ provided by Tableau were used for its creation. The second graph showed provinces taking into consideration the maximum price and minimum price of the wines. The third graph displayed the variety and the exact marked price across the province of the particular country. These second and third graphs were created by the use of ‘Packed Bubbles’ from Tableau. Appropriate conclusions can be made by playing around with visualisations with respect to the three main features of the wine namely, variety, price and points(ratings).

Web Application Development and Deployment

Frontend: We have used HTML, CSS and Bootstrap to design the webpage. Also, JQuery and JavaScript has been used to embed Tableau Dashboard and make the WebApp interactive. There are three options present on the navigation bar i.e. Home, Countries and Visualizations. Home tab shows the basic introduction of the App and Dataset. Countries tab show the list of countries with their contribution to Wine production. Visualisation tab shows the embedded Tableau Dashboard which can be further used to apply filters on visualizations.

Backend: The backend of the application was developed using microframework Flask, based on Python. Furthermore, Tableau was integrated into the application using Tableau JS-API to demonstrate the analytics. The app was containerized using Docker on Google Cloud Platform. After the image was created, it was stored in Google Container Registry (GCR). A Kubernetes cluster of size 3 was developed. The image from GCR was deployed, as a Kubernetes service using LoadBalancer on the external IP http://35.202.230.217/

Challenges faced:

1. The main challenge we faced was the integration of Tableau with Python. There is a library in Python named TabPy, which we hoped to use in the initial stages. But as we progressed, it became difficult to integrate the Tableau to backend as TabPy is still missing features. Hence, we ended up using the JS API for Tableau.
2. Another challenge we faced was the use of PIG/Hive and their integration with Flask. These technologies are primitive as compared to their counterpart like Apache Spark, when it comes to integration.

Lessons Learned:

1. Using lightweight applications and API is beneficial in terms of cost and time efficiency.

Individual Contributions

The team members were not just limited to the tasks mentioned below. They learnt and grew by helping each other out wherever it was necessary.

1. **Bhargav :**
    - Loading dataset into HDFS
    - Creating dataframes in SPARK
    - Storing data from HDFS to SPARK

2. **Yashaswi:**
    - Cleaning dataset using SPARK(Scala)
    - Creating bucket in Big Query
    - Connection of Big Query in Tableau.

3. **Nikhil:**
    - Created a dataflow diagram for the App Construction that explained several steps of the project.
    - Created a visualization of World Map based on average points and average Price of the wines across different provinces in the countries taken.

4. **Alekhya:**
    - Implementing the UI of the application by means of HTML, CSS, Bootstrap and JQuery.
    - Displaying tableau data in appropriate visualisations on the webapp.
    - Created the 'Packed Bubbled' graphs for identification of wines across different countries by
        a. Maximum and minimum prices of the wine .
        b. Wine Variety and Price

5. **Akanksha:**
    - Integration of tableau dashboard.
    - Making web app interactive with JQuery
    - Styling of the Web app
    - Connecting database to Tableau server.

6. **Joseph**
    - Designed the User Interface.
    - Assisted with the development of User Interface Development using HTML,CSS and JavaScript

7. **Himanshu**
    - Tokenized the description while cleaning the dataset.
    - Helped with streamlining the bucket to BigQuery
    - Used Python 3.7, along with microframework Flask to build the application and integrate everything.
    - Containerized the web application using Docker
    - Deployed the docker image via Kubernetes

Student ID	Name	Role	Contribution
19210249	Himanshu Vashisht (GC)	Backend/Deployment	Satisfactory
19210509	Nikhil Mittal	Visualization	Satisfactory
19211007	Yashaswi Verma	Data Cleaning / Loading	Satisfactory
19210256	Alekhya Nethra	Visualization	Satisfactory
19210189	Bhargav Dandu	Data Cleaning / Loading	Satisfactory
19210222	Joseph Cherian	Front-end	Satisfactory
19210135	Akanksha Rajpute	Front-end	Satisfactory

References:

[1] "Wine Reviews | Kaggle." [Online]. Available: https://www.kaggle.com/zynicide/wine-reviews. [Accessed: 15-Nov-2019].

[2] “Buckets | Cloud Storage | Google Cloud.” [Online]. Available: https://cloud.google.com/storage/docs/json_api/v1/buckets. [Accessed: 20-Nov-2019].
[3] “The Benefits of Combining Google BigQuery and BI | Sisense.” [Online]. Available: https://www.sisense.com/blog/the-benefits-of-combining-google-bigquery-and-bi/. [Accessed: 20-Nov-2019].
[4] “Advantages and Disadvantages of Tableau - AbsentData.” [Online]. Available: https://www.absentdata.com/advantages-and-disadvantages-of-tableau/. [Accessed: 25-Nov-2019].
[5] “Welcome to Flask — Flask Documentation (1.1.x).” [Online]. Available: http://flask.palletsprojects.com/en/1.1.x/. [Accessed: 01-Dec-2019].
[6] “Docker Documentation | Docker Documentation.” [Online]. Available: https://docs.docker.com/. [Accessed: 02-Dec-2019].
[7] “Tutorials - Kubernetes.” [Online]. Available: https://kubernetes.io/docs/tutorials/. [Accessed: 03-Dec-2019].