CLOUD TECHNOLOGIES

CA675

Assignment-2 Cloud Application

Group-O Report

We declare that this material, which we now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. We understand that plagiarism, collusion, and copying are grave and serious offenses in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations set out in the module documentation. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged, and the source cited are identified in the assignment references.
This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found recommended in the assignment guidelines.

Name	Student Id	E-mail Id
Aditya Kumar Yadav	18210034	Aditya.yadav3@mail.dcu.ie
Alekhya	18210618	Alekhya.singh7@mail.dcu.ie
Kashyap Krishnamurthy	18210248	kashyap.krishnamurthy2@mail.dcu.ie
Paritosh Gupta	18210686	Paritosh.gupta3@mail.dcu.ie

Date: April 19, 2019

Table of Contents

Productive Reviews Affirmation App

1. Introduction

The advancements in Cloud Computing has driven the explosion of e-commerce; it has become the preferred means of purchasing certain products and services. An integral part of a product/service listed on websites is the 'Review'. Customer reviews are very important to users and the service providers. Reviews form a crucial part in the decision-making process for a person intending on making the said purchase. Statistics suggest that 90% of customers read reviews online before visiting a business. However, not all reviews prove to be helpful - ensuring a review is 'Productive' would help customers and businesses immensely.

The Productive Reviews Affirmation Application (PRAA) focuses on enriching the review writing process. This application augments the reviewer using Machine Learning (ML). The application helps in validating the helpfulness of review provided by the user. The ML system has been trained on existing reviews, it then processes a review and encourages the composer to provide additional details, should there be a need.

2. Planning

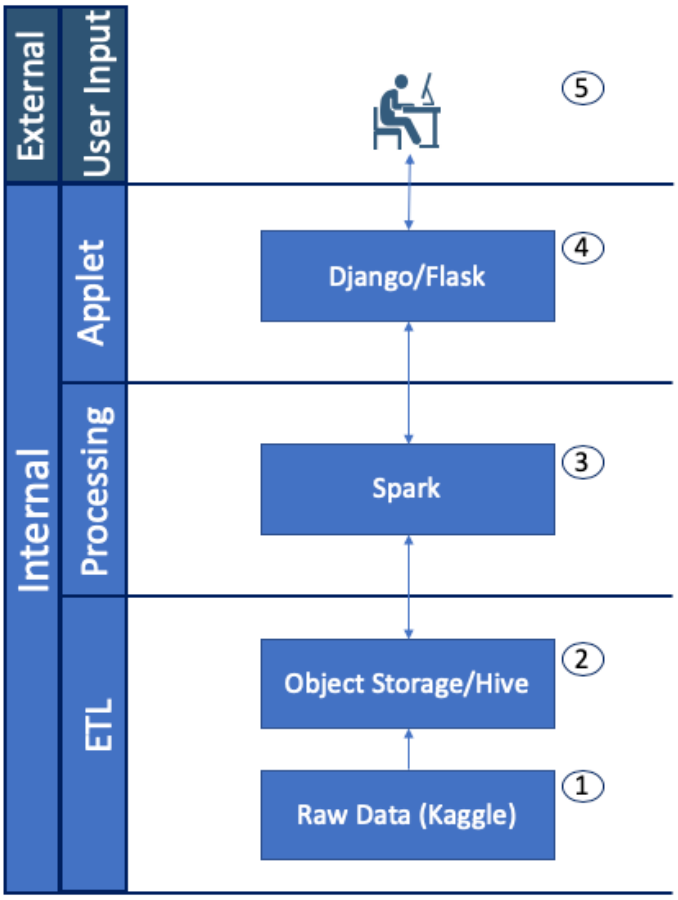
This section details the various aspects of the project planning phase.

2.1 Dataset

The dataset chosen for the solution consists of reviews for fine-foods from Amazon and other Amazon categories. The data, sourced from Kaggle, spans a period of more than 10 years. The data consists of over 560,000 reviews up to October 2012. Link to the dataset on Kaggle – click here.
Records include –

Id	-	*Record Id*
ProductIdUnique	-	*identifier for the product*
UserIdUnqiue	-	*identifier for the user*
ProfileNameProfile	-	*name of the user*
HelpfulnessNumeratorNumber	-	*users who found the review helpful*
HelpfulnessDenominatorNumber	-	*users who indicated if the review was helpful*
ScoreRating	-	*between 1 and 5*
TimeTimestamp	-	*time of the review*
SummaryBrief	-	*summary of the review*
TextText	-	*the review body*

2.2 Solution Design



2.3 Key Technologies

1. Raw data from Kaggle
2. Object Storage/Hive
3. Apache Spark
4. Flask with HTML/CSS/Javascript

2.4 Take away from Mid-Way Report Feedback

From feedback provided in the midway report, we implemented the frontend using Flask instead of Django. Flask is lightweight, flexible and has the third-party libraries which makes the implementation effective.

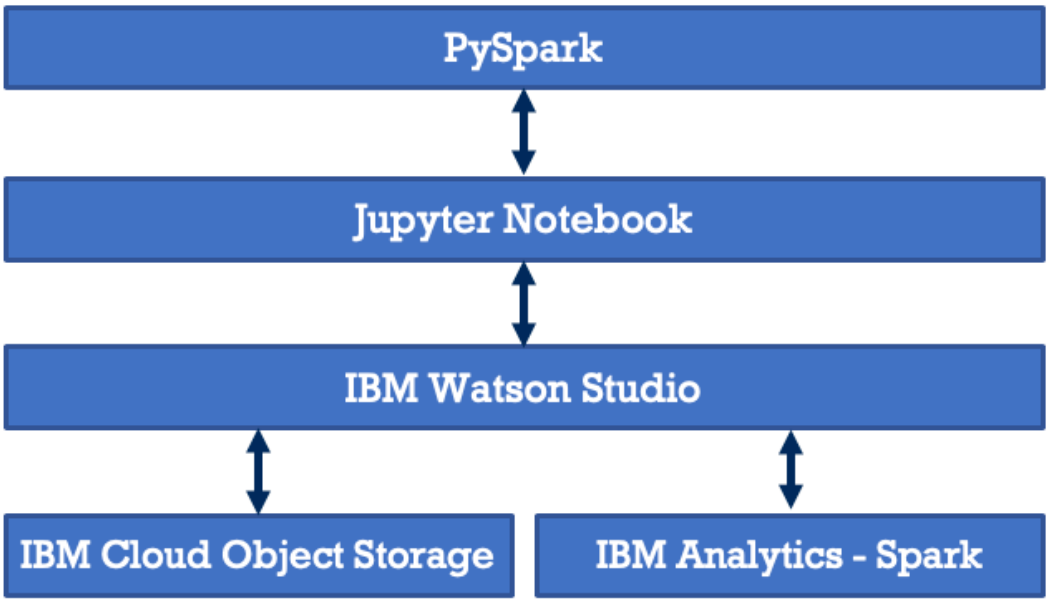
3. Implementation

The following section details the PRAA solution implementation.

3.1 Solution Outline

The solution has been implemented in two stages – 1. IBM Cloud and 2. Local (laptop)
This has been done due to the limited amount of free compute available on the IBM Cloud. Hence, the compute intensive tasks of data cleaning, training and testing the ML algorithms have been performed on the cloud.

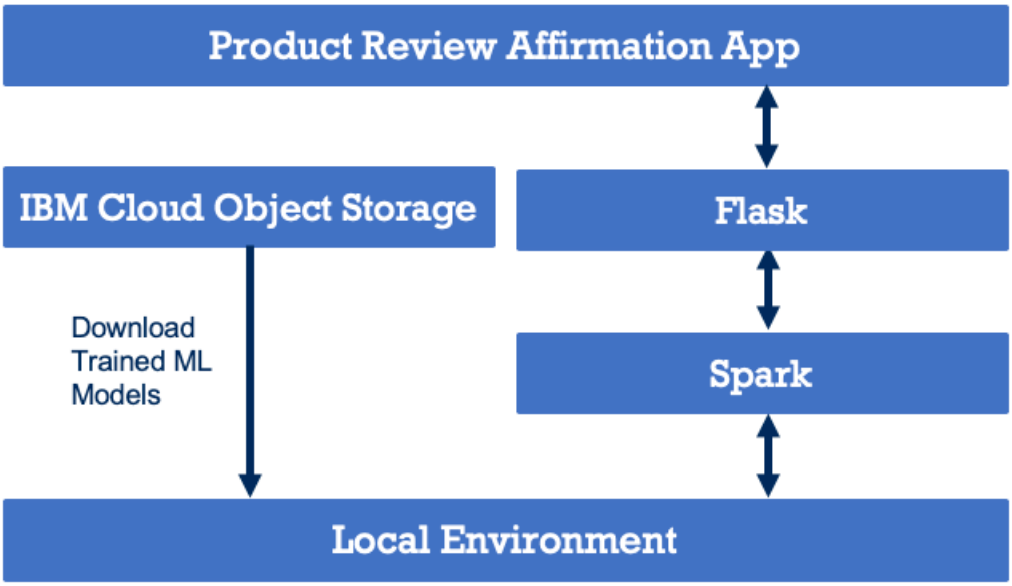
Stage 1 : Development – The development of the solution has been performed on IBM Cloud. Two services for storage and compute have been created, Object Storage and Spark, respectively. These services are integrated with IBM Watson Studio, this platform provides the functionality to create projects and have runtimes associated with each project. Jupyter Notebook for PySpark has been created in this project space. The notebook uses Spark as the runtime kernel. The Object storage has been configured with a dedicated bucket. All the data associated such as the dataset, the notebook exports and logs are stored on this storage bucket.



Build Stack –

- Object Storage - 1 S3 bucket
- Spark v2.3 - Driver: 1vCPU, 4 GB RAM
  Executor: 1 executor – 1vCPU, 4GB RAM
- Jupyter Nb. - Python v3.5

Stage 2: Deployment – Once the desired ML models have been created, the models are written to Object Storage. The model is then downloaded from the cloud and deployed locally.



Build Stack –

- Spark v2.3
- Anaconda - Python 3.7
  - Flask 1.0.2

3.2 Implementation Steps

1. Data collection and storage -
   a. The dataset is over 568,0454 Amazon Fine Food Reviews scrapped from the website. This has been downloaded from Kaggle.
   b. Collected data is stored in the IBM Cloud – Object Storage.

2. Machine learning model using PySpark on IBM Watson studio -
   a. The stored data is then processed in a Spark runtime where the data has been wrangled, modeled and fitted into a chosen ML Algorithm. This ML model then analyzes new text(reviews) and ascertains if the review is useful or not based on the training data.
   b. The final models are then written back to Object Storage, from where it is downloaded.

3. Local environment –
    a. Spark has been installed and configured on macOS Mojave.
    b. The locally running Spark has been integrated with Anaconda such that the Web App created using Flask can be run using Spark as the Kernel.

4. Web App and ML integration using Flask framework -
    a. We have developed web-app in flask framework, where '*user review form*' with necessary field level validation has been created which can be easily linked with any different web app in future.
    b. Above developed Machine Learning model is downloaded from cloud and is integrated with user review form and web page in the flask app main file to make it as one whole fully functional app.

5. Using Spark submit, the flask web server was started, and the web app was accessible through local host on port 8081. The input review string from the user is first converted into data frame, tokenized and analysed using the ML model for the usefulness prediction.

4. Performance

Performance of the system is based on the accuracy of the model and how efficiently the system provides the helpfulness of the reviews. The reviews entered by the user as input should give a result that shows whether the entered text is a useful review or need to add more details. After testing the model with 15% of the data, the system reached an accuracy of 83% (on a certain randomized seed).

5. Resources

- The notebook on IBM Watson can be accessed by following this link. (IBM ID is however required to run the code)
- The codes and documentation can be found in this Git repo.

6. Conclusion

Rapid evolution of Cloud Technologies has driven the boom of the online market. Due to the increased demand for online capabilities, a critical need for research in this important area is needed. Poor quality reviews hurt business as these reviews are the major reason that helps people to buy goods online. Reviews substitute the in-person analyses of a given product prior to making a purchase. Access to other people's opinions is crucial. In this project we have built a viable solution to ensure Reviews online are Productive.

7. Contribution

All the team members have contributed equally in all the areas of the solution. The table presents the member who shouldered a given functionality.

Pipeline	Responsibility
Storage – IBM Object Storage	Aditya, Kashyap
Processing Engine- Spark (PySpark)	Kashyap, Alekhya
User Interface	Alekhya, Paritosh
Integration Services	Aditya, Paritosh, Kashyap

Below is the percentage contribution for the individuals.

Team Member	% Contribution
Aditya Kumar Yadav	25%
Alekhya	25%
Kashyap Krishnamurthy	25%
Paritosh Gupta	25%