**Productive Reviews Affirmation App**

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:**

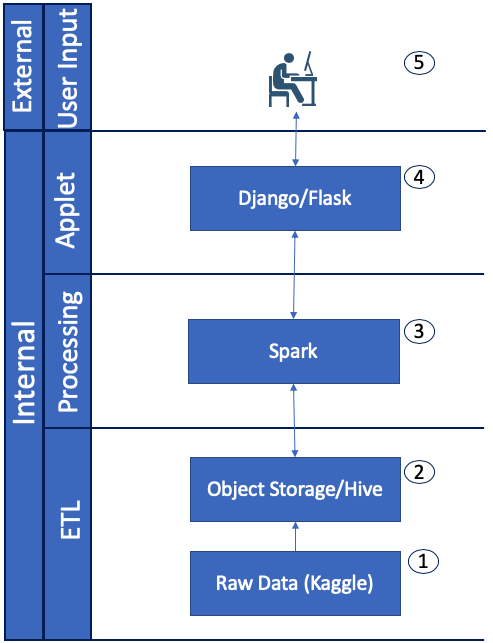
# **Introduction:**

Customer reviews are very important to users and service provider. The reviews always have a tremendous impact on making better decisions regarding the purchase. There is a fact about consumers that 90% of them read online reviews before visiting a business. If you get the affirmation on those reviews whether it would be useful to others or not, that would be like a cherry on the cake.

The Productive Reviews Affirmation Application (PRAA) focuses on enriching the review writing process. This application basically helps in validating the helpfulness of review provided by the user to others. From the past reviews, we have built a machine learning model that verifies the reviews and encourage the users to provide helpful reviews.

# **Planning:**

**Pipeline Flowchart:**



## **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 – <https://www.kaggle.com/snap/amazon-fine-food-reviews>.

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* |

## **Key Technologies:**

* Raw data from Kaggle
* Object Storage/Hive
* Apache Spark
* Flask
* HTML/CSS/Javascript

## **Take away from Mid-Way Report Feedback:**

From feedback provided for the midway report, we implemented frontend using Flask instead of Django due to its facilities like lightweight, flexibility and has the third-party libraries which make the implementation simple.

# **Implementation:**

## **Steps followed:**

1. Data collection and storage -
   1. Data which comes from over 568,0454 Amazon Fine Food Reviews is collected from Kaggle.
   2. Collected data is stored in an Object Storage, IBM Watson Studio.
2. Machine learning model using PySpark -
   1. The stored data is then processed in a Spark runtime where the data got wrangled, modeled and fitted into a chosen ML Algorithm. This ML model then analyzes new text(reviews) and ascertain if the review is useful or not based on the training data.
3. Web App and ML integration using Flask framework -
   1. 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.
   2. Above developed Machine Learning model, user review form and web page has been integrated in the flask app main file to make it as one whole fully functional app.
   3. Using PySpark submit, the flask web server was started, and the web app was accessible through local host link. The input review string from the user is first converted into data frame, tokenized and called ML model for the usefulness prediction.
4. Final deployment on IBM Watson studio -

# **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%.

# **Conclusion:**

As we all know, the technologies are evolving every day and with the development of these technologies, online shopping has also boomed the market with time. Due to the increased demands for online shopping, 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 without seeing the items themselves because they get access to other people’s opinions for that respective item. So, in this project, we have not focused on the Score, but only the positive/negative sentiment of the recommendation.

Finally, we have tried to find the "best" model for predicting the recommendation's sentiment.

# **Contributions:**

|  |  |
| --- | --- |
| **Pipeline** | **Primary Responsibility** |
| Raw Data Ingest – Hive/Object Storage | Aditya, Kashyap |
| Processing Engine- Spark (PySpark) | Kashyap, Alekhya |
| User Interface | Alekhya, Paritosh |
| Integration Services | Aditya, Paritosh |