**Project Proposal Draft for**

Sales Analyzer

Group – 6

**Group Members:** Arpan Sharma

Sarath Kalaga

Himanshu Sharma

Teja Tirunagari

# Description:

We want to develop a system application for Sales representative of a company. The primary aim of this application is to perform sales analysis on reviews of different products by opinion mining. Find out patterns which may lead us to find products that are bought together. Sales representative use this application to analyze these reviews and easily dissect what people praise or protest about their experience with the product. This can also estimate how well the product is responding in the market, which are of market is it having a favorable response and in which a negative response. We believe that it is important to classify incoming customer reviews about a product based on key aspects of a brand’s product that customers care about and, the sales manager's underlying intentions and reactions concerning those aspects. When these basic concepts used in combination, become a very important tool for analyzing the customers feeling towards different products or brands and why consumers don’t buy some products.

**Dataset:** The dataset contains basic product information about customer id, rating, review and more features for each product listed on Amazon website.

**Preprocessing:** Each review of the different product will be tokenized, and POS tagging will be performed.

**Opinion Mining:** After preprocessing, the adjectives from the review are compared to opinion lexicons. All the reviews are summarized according to positive or negative lexicons in the review after this step.

**Pattern Mining:** Pattern like how likely a customer is going to buy two different products together or how likely is a customer going to buy a product will be explored.

**Visualization:** The patterns, sales figures and analysis will be visualized for an easy understanding for the sales representative.

**Language used:** Python 3

**Platform:** Anaconda (Jupyter Notebooks),

**GUI:** In this project we would be using tkinter package for developing GUI. Animations are also managed using tkinter package. Fastest and easiest way to create GUI can be done using tkinter. tkinter is now included in any Python distribution. No supplementary modules are required in order to run scripts using tkinter.

# 

# Directory Structure:

**a. GUI**

**b. Dataset**

**c. Preprocessing**

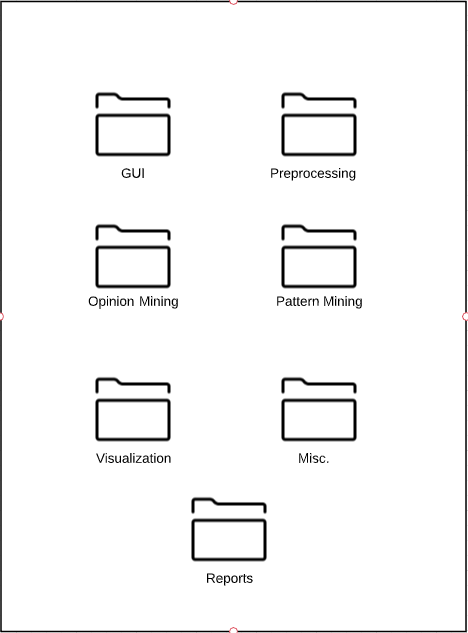
**d. Opinion Mining**

**e. Pattern Mining**

**f. Visualization**

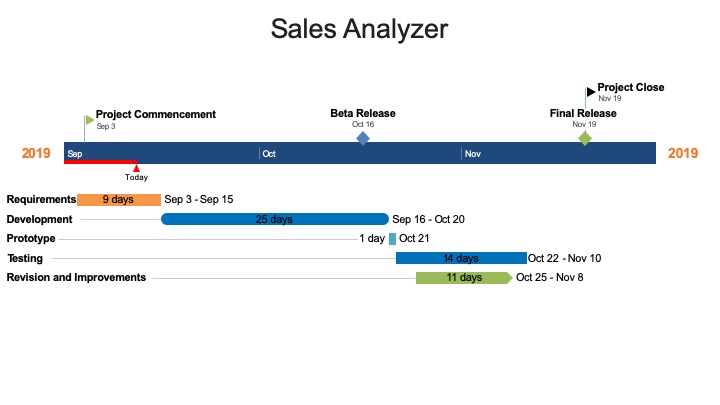
**g. Miscellaneous**

**h. Reports**

****

Code and different files are categorized according to above following directories to keep it simple and organised.All the deliverables and presentations will be uploaded in Miscellaneous.

# Gantt/PERT Chart

**Risk Management:**

The process of risk identification  and designing a contingency plan is an essential part of a project. We have decided to follow a procedure of maintaining a “Risk\_Archive” which is a very simple excel workbook in order to keep a record of each identified risk, it’s detailed description, probability of occuring, it’s impact and the current status. This initiative would not only help us to make it simpler for team members to coordinate the details of every risk amongst the team but also would make risk management a vital component of our entire process of project development.

The diagram below is a snapshot of this file.

**A screenshot of a cell phone

Description automatically generated**

The risks identified by our team so far are as follows:

1. **Dataset:**

One of the early steps in our development phase is to start working upon and preprocessing on a dataset in order to cleanse the data to be used in the next phase of mining. But, the primary issue we are facing at the moment is that the dataset selected at the moment only consists of features which could be utilized in order to train a model for mining opinions to visualize performance of a product based on the opinions but the current dataset does not have sufficient features which would enable us to mine association rules.

**Contingency Plan**: Acquire the desired values or features missing in the dataset from the internet by writing a code. Or choose a different dataset. Or Work with the dataset that you have and look for different patterns instead of different datasets.

1. **Integration of modules:**

Integration of GUI and backend code is a challenging task. Working on the code with opinion mining , converting the output and displaying the results in an understandable and visualised way might be challenging. Updating the GUI parallely with the backend can cause unknown risk

**Contingency plan:** Perform periodical check along the way with project progress.

1. **Visualization:**

Visualizing in a python code can be easily executed, but displaying it in a high resolution in the GUI is a bit challenging. The visualization might deviate from results because the model might not be 100% accurate.

**Contingency plan:** It can be avoided or reduced with proper model management such as testing, governance policies and independent review.

1. **Project Schedule:**

Although project schedule does not possess much risks. Any emergencies to group members may delay or push the intended deadlines.

**Contingency Plan**: Communicate with the team. Inform everyone and plan accordingly.

# **Roles and Responsibility:**

The roles and responsibilities have been designated as follows:

**a.** **GUI:**

o  Arpan Sharma

o  Sarath Kalaga

**b.**   **Dataset:**

o  Himanshu Sharma

o  Arpan Sharma

**c.** **Preprocessing:**

o  Sarath Kalaga

o  Himanshu Sharma

**d.**   **Opinion Mining:**

o  Himanshu Sharma

o  Teja Tirunagari

**e.**   **Association Analysis:**

o  Teja Tirunagari

o  Arpan Sharma

**f.** **Visualization:**

o  Sarath Kalaga

o  Teja Tirunagari

The project members would be working primarily onto their assigned modules. At regular weekly intervals, team meetings have been scheduled twice every week in order to report the progress on each module as well as to discuss any new identified risks and the status of current risks. Each developer is supposed to update and/or check the risk archive at regular intervals in order to monitor them and thus, ensure efficient project development and deployment.

Without messing master each project member will create their own branch update code and Once the code is updated project member would pull request notifying other members in the project that they need to review the code and merge into the master.