CHAPTER-1
INTRODUCTION

In today's fast-paced and interconnected global marketplace, businesses face an unprecedented challenge: the constant scrutiny of their products and reputation by consumers, competitors, and stakeholders. The success of any organization hinges not only on the quality and performance of its products but also on its ability to effectively manage its reputation in the digital age. This project delves into the multifaceted world of product monitoring and reputation management, two interrelated facets of contemporary business strategy that are pivotal for organizational growth and sustainability.

1.1 Background:

The digital revolution has transformed the way businesses operate, communicate, and compete. The advent of the internet and social media has given consumers unparalleled access to information, enabling them to scrutinize products and services like never before. Simultaneously, organizations are challenged to navigate an intricate web of public perception, where reputation is currency, and a single viral post can shape public opinion in a matter of hours.

In this landscape, a company's products and its reputation are inextricably linked. Product quality directly impacts brand perception, customer loyalty, and market share. Conversely, a company's reputation can significantly influence consumer trust, investor confidence, and stakeholder relationships. Therefore, effective product monitoring and reputation management have emerged as essential components of modern business strategy.

1.2 Objectives of the Project:

The primary objective of this project is to comprehensively explore the concepts, methodologies, and strategies associated with product monitoring and reputation management. Specifically, the project aims to analyse the theoretical foundations and practical implications of product monitoring and reputation management in contemporary business practices.

Examine the methods and tools employed by organizations to monitor product performance, gather customer feedback, and ensure the continuous improvement of their offerings. Investigate the strategies and tactics used to manage and enhance reputation, both proactively and reactively, in response to public sentiment and crises. Provide a thorough examination of real-world case studies that highlight successful approaches to product monitoring and reputation management across diverse industries.

1.3 Scope of the Project

This project will encompass a wide array of industries, from consumer goods and technology to healthcare and finance, to showcase the universal relevance of product monitoring and reputation management. It will also explore the role of digital technology, social media, and data analytics in shaping these practices.

Through a combination of literature review, case studies, and data analysis, this project will offer insights and recommendations to help businesses navigate the complexities of product monitoring and reputation management. Furthermore, it will underscore the critical importance of these two facets in building and sustaining trust among consumers, stakeholders, and the broader public.

CHAPTER 2
LITERATURE SURVEY

1. "Irina Dadova, Jakub Soviar", "University of Zilina, 2022", "The recent theoretical framework for online reputation management".

Summary: The article discusses why it is necessary to manage the online reputation of a company and offers a basic theoretical basis for reputation management.

2. "Tao Chen, Premaratne Samaranayake, Xiong Ying Cen, Meng Qi, Yi Chen Lan", "Frontiers in Psychology, 2022", "The impact of online reviews on consumers' purchasing decisions".

Summary: This study investigated the impact of online product reviews on consumers purchasing decisions by using eye-tracking.

3. "Dong Zhang, Wenwen Li, Baozhuang Niu, Chong Wu", "IEEE Transactions on Knowledge and Data Engineering, 2022", "A machine learning approach to fake product review detection".

Summary: End-to-end framework to detect fake reviewers based on behaviour an textual information.

4. "J. M. Alsaraireh, N. A. Shamaileh, S. Saraireh, M. K. Al-Azzam, R. K. Kanaan, A. Mohammad, S. I. Al-Hawary", "Journal of Business Strategy, 2022", "The impact of online reviews on brand equity".

Summary: Brand value is obtained through surveys to understand the overall mental image of the brand.

5. "Syed Mohammed Anas, Santoshi Kumari", "ICICT 2021", "Opinion mining based fake product review monitoring and removal system".

Summary: Fake review detection and its elimination from the given dataset using different Natural Language Processing (NLP) techniques is important in several aspects.

6. DergiPark", "Dogus University Dergisi, 2021" ,"A review on online reputation management and online reputation components".

Summary: Study to define the components that create, develop and maintain a robust reputation in the virtual world, examine the relationship between these components and explain their functions.

7. "Ata-Ur-Rehman, Nazir M. Danish, Sarfraz M. Tanzeel, Nasir Usama", "CCE 2019", "Intelligent interface for fake product review monitoring and removal".

Summary: An Intelligent Interface takes the Uniform Resource Locator (URL) related to products of Amazon, Flipkart and analyses the reviews, and provides the customer with the original rating.

8. "Raheesa Safrin, K.R. Sharmila, T.S. Shri Subangi, E.A. Vimal", International Research Journal of Engineering and Technology IRJET 2017", "Sentiment analysis on online product reviews".

Summary: Sentiment classification is used to verify or analyze the comments given by the user to extract the opinion from it.

9. "Pranali Borele, Dilipkumar A. Borikar", "IOSR Journal of Computer Engineering, IOSR-JCE 2021", "An approach to sentiment analysis using artificial neural network with comparative analysis of different techniques".

Summary: Machine learning-based approaches to sentiment analysis and brings out the salient features of techniques in place.

10. "Xing Fang and Justin Zhan", "Journal of Big Data 2, 2022", **"**Sentiment analysis using product review data - Fang and Zhan Journal of Big Data".

Summary: Hybrid approaches use machine learning to predict the sentiment polarity of a review or sentence, and then use lexicon-based methods to refine the prediction.

CHAPTER 3
METHODOLOGY

This chapter outlines the comprehensive methodology employed to investigate the intricate domains of product monitoring and reputation management. The chosen methodology combines qualitative and quantitative research methods to ensure a thorough examination of these critical aspects of contemporary business strategy.

3.1 Research Design

A mixed-methods research design was selected to provide a holistic understanding of product monitoring and reputation management. This approach integrates both qualitative and quantitative data collection and analysis techniques, allowing for a multidimensional exploration of the research objectives.

3.2.1 Machine learning:

Machine learning is a branch of artificial intelligence(AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy. Machine learning (ML) plays a significant role in product monitoring and reputation management by automating and enhancing various aspects of these processes. Machine learning is an important component of the growing field of data science. Through the use of statistical methods, algorithms are trained to make classifications or predictions, and to uncover key insights in data mining projects. These insights subsequently drive decision making within applications and businesses, ideally impacting key growth metrics. As big data continues to expand and grow, the market demand for data scientists will increase. They will be required to help identify the most relevant business questions and the data to answer them.

Sentiment Analysis:

Product Monitoring: ML algorithms can be used to analyze customer feedback, reviews, and social media mentions related to a product. Sentiment analysis models classify text as positive, negative, or neutral, allowing companies to gauge customer sentiment and identify areas for product improvement.

Reputation Management: Sentiment analysis helps organizations monitor online conversations about their brand and products. By automatically tracking sentiment, companies can respond quickly to negative comments or trends and take actions to protect their reputation.

Review Summarization:

Product Monitoring: ML models can summarize lengthy product reviews to provide a concise overview of customer opinions and pain points. This helps product teams quickly identify common issues and prioritize improvements.

Recommendation Systems:

Product Monitoring: Recommendation algorithms can suggest related products or upgrades based on customer preferences and purchase history, thereby enhancing the customer experience and increasing sales.

Customer Segmentation:

Reputation Management: ML can segment customers into different groups based on their behaviour and preferences. For example, identifying high-value customers allows companies to prioritize addressing their concerns and ensuring a positive experience.

Predictive Analytics:

Product Monitoring: ML models can predict product defects or issues before they become widespread, enabling proactive maintenance and quality control.

Reputation Management: Predictive models can forecast potential reputation risks by analyzing historical data and emerging trends. This allows companies to take preventive measures and respond quickly to potential crises.

Social Media Monitoring:

Reputation Management: ML-powered social media monitoring tools can track brand mentions, trends, and sentiments across various platforms. These tools can automatically alert companies to any unusual spikes in activity, allowing for timely responses.

Content Moderation:

Reputation Management: ML models can help filter and moderate user-generated content to prevent harmful or inappropriate content from tarnishing a brand's reputation.

Image and Video Analysis:

Product Monitoring: ML can be used to analyze images and videos related to products. For instance, image recognition can identify product defects in manufacturing processes.

Anomaly Detection:

Reputation Management: ML models can detect unusual online behavior or patterns that may indicate a reputation-threatening event, such as a sudden surge in negative reviews or a coordinated social media attack.

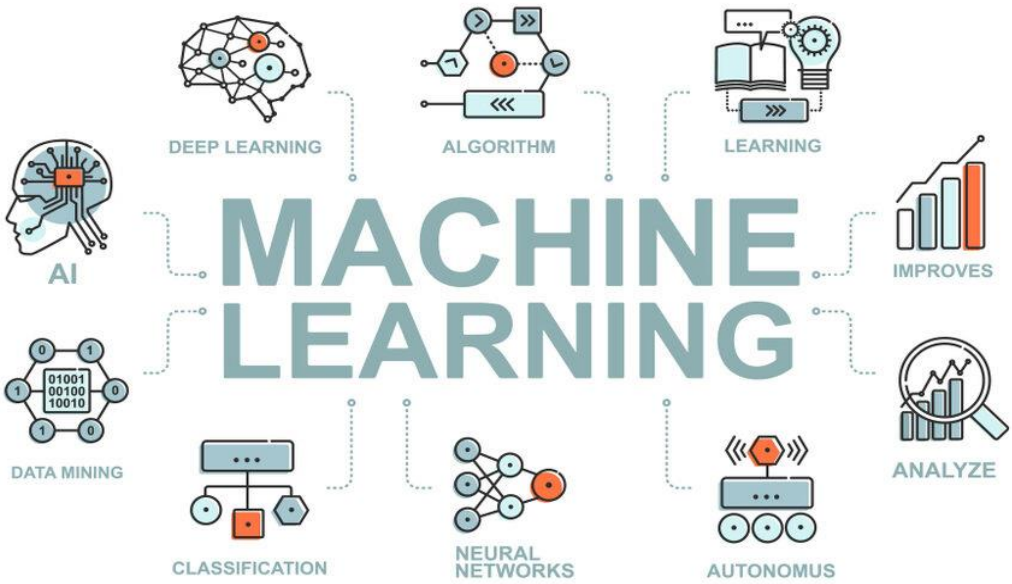
Natural Language Processing (NLP):

Product Monitoring: ML-powered NLP can extract valuable information from unstructured text data, such as customer feedback, warranty claims, or support tickets, to identify product issues and areas for improvement.

Reputation Management: NLP techniques can be used to analyze and categorize text data from various sources, helping organizations gain deeper insights into customer opinions and sentiments.

Continuous Improvement:

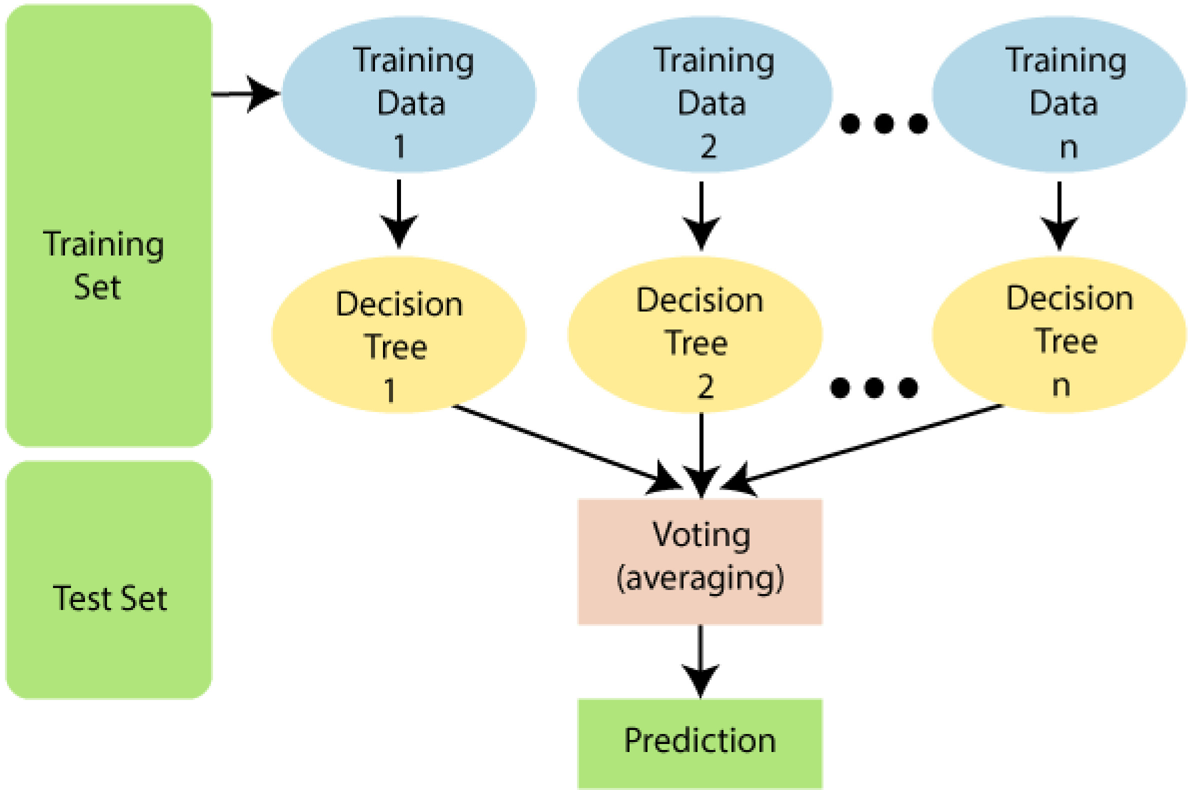
Product Monitoring: ML models can continuously learn from new data, enabling product teams to adapt and improve products based on evolving customer feedback and market trends.



3.2.2 Random Forest Algorithm:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result. The predictions from each tree must have very low correlations.



3.2.3 Page rank algorithm:

PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:

"PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites."

It is not the only algorithm used by Google to order search engine results, but it is the first algorithm that was used by the company, and it is the best-known.

Algorithm:

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called “iterations”, through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

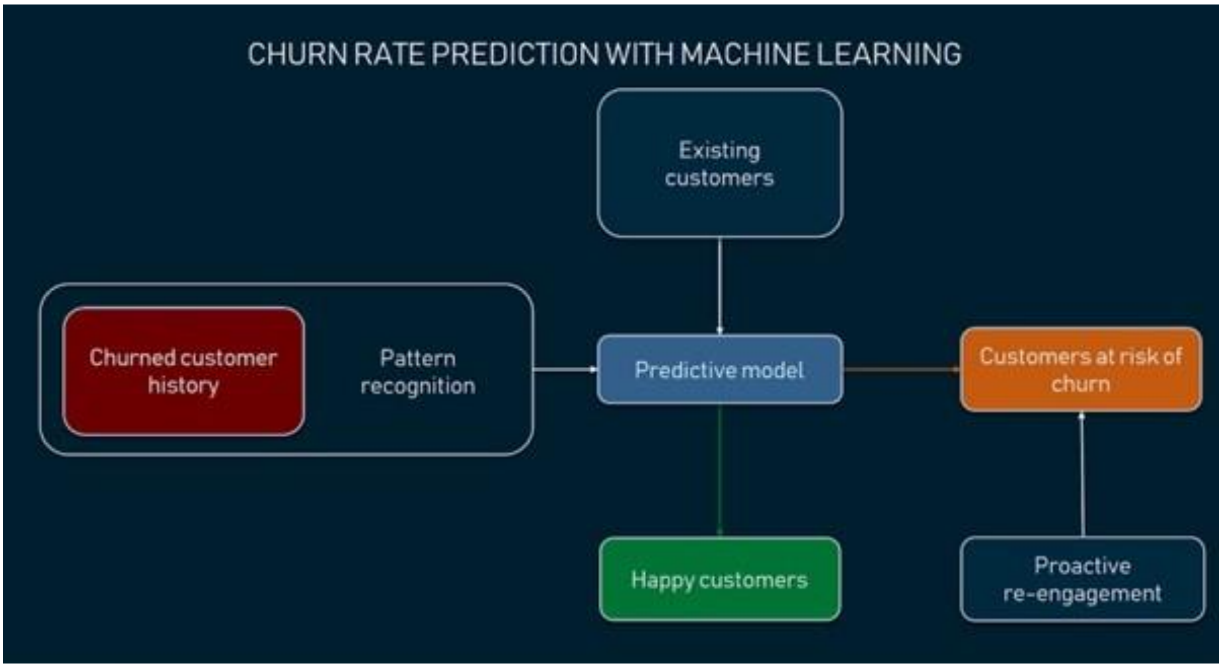
3.2.4 Customer Churn Prediction:

Customer churn (or customer attrition) is a tendency of customers to abandon a brand and stop being a paying client of a particular business. The percentage of customers that discontinue using a company’s products or services during a particular time period is called a customer churn (attrition) rate. One of the ways to calculate a churn rate is to divide the number of customers lost during a given time interval by the number of acquired customers, and then multiply that number by 100 percent. For example, if you got 150 customers and lost three last month, then your monthly churn rate is 2 percent.

Churn rate is a health indicator for businesses whose customers are subscribers and paying for services on a recurring basis, notes head of data analytics department at ScienceSoft Alex Bekker, “Customers [of subscription-driven businesses] opt for a product or a service for a particular period, which can be rather short – say, a month. Thus, a customer stays open for more interesting or advantageous offers. Plus, each time their current commitment ends, customers have a chance to reconsider and choose not to continue with the company. Of course, some natural churn is inevitable, and the figure differs from industry to industry. But having a higher churn figure than that is a definite sign that a business is doing something wrong.”

There are many things brands may do wrong, from complicated onboarding when customers aren’t given easy-to-understand information about product usage and its capabilities to poor communication, e.g. the lack of feedback or delayed answers to queries. Another situation: Longtime clients may feel unappreciated because they don’t get as many bonuses as the new ones. In general, it’s the overall customer experience that defines brand perception and influences how customers recognize value for money of products or services they use.

The reality is that even loyal customers won’t tolerate a brand if they’ve had one or several issues with it. For instance, 59 percent of US respondents to the survey by PricewaterhouseCoopers (PwC) noted that they will say goodbye to a brand after several bad experiences, and 17 percent of them after just one bad experience.



3.3 Ethical Considerations

Throughout the research process, ethical considerations were upheld. Informed consent was obtained from participants in interviews and surveys, and steps were taken to ensure the confidentiality and anonymity of respondents. Additionally, all data sources, including secondary data from literature and online platforms, were appropriately cited and referenced.

3.4 Tools and Software Used

Various tools and software were employed to facilitate data collection and analysis, including statistical analysis software, content analysis tools, data visualization software, and survey software.

CHAPTER 4
DESCRIPITION OF PROPOSED SYSTEM

Designing a comprehensive model for product monitoring and reputation management involves a multifaceted approach that integrates various methodologies, technologies, and strategies. In this proposed model, we'll delve into the intricacies of product monitoring and reputation management, discussing the importance, challenges, key components, and implementation strategies. This model aims to provide organizations with a robust framework to effectively monitor their products across different channels and manage their online reputation proactively.

4.1 MONITORING TOOLS AND TECHNOLOGIES

4.1.1 Social Media Listening Tools

Social media platforms serve as valuable sources of consumer sentiment and feedback. Social media listening tools enable organizations to monitor conversations, mentions, and trends related to their products or brand. Key features of these tools include:

Real-time monitoring:

Continuous tracking of social media channels for mentions of the brand or specific products.

Sentiment analysis:

Automated analysis of the sentiment associated with mentions (positive, negative, or neutral).

Trend analysis:

Identification of emerging trends and topics relevant to the brand or industry.

Engagement metrics:

Metrics such as likes, shares, and comments to gauge the level of audience engagement.

Popular social media listening tools include:

- Brandwatch
- Sprout Social
- Hootsuite
- Mention
- Talkwalker

4.1.2. Review Monitoring Platforms

Review monitoring platforms focus on aggregating and analyzing customer reviews from various online platforms, including e-commerce websites, review sites, and forums. These platforms offer features such as:

Centralized dashboard:

A unified interface for monitoring reviews across multiple platforms.

Review analysis:

Sentiment analysis and categorization of reviews based on themes or topics.

Review response management:

Tools for responding to customer reviews directly from the platform.

Competitor analysis:

Comparison of product reviews with competitors' offerings.

Prominent review monitoring platforms include:

- Trustpilot
- Yelp for Business
- Google My Business
- TripAdvisor for Business
- Amazon Seller Central

4.1.3. Web Scraping Techniques

Web scraping involves extracting data from websites to gather information relevant to product monitoring and reputation management. While manual web scraping is an option, automated web scraping tools offer efficiency and scalability. Key features of web scraping tools include:

Customizable scraping parameters:

Ability to specify the data fields and sources to be scraped.

Scheduled scraping:

Automation of regular data extraction tasks at predefined intervals.

Data cleansing:

Filtering and preprocessing of scraped data to remove irrelevant or duplicate information.

Integration capabilities:

Exporting scraped data to other tools or platforms for further analysis.

Popular web scraping tools and frameworks include:

- BeautifulSoup (Python library)
- Scrapy (Python framework)
- Octoparse
- Import.io
- ParseHub

4.1.4. Sentiment Analysis Software

Sentiment analysis software utilizes natural language processing (NLP) and machine learning algorithms to analyze textual data and determine the sentiment expressed within it. These tools can be integrated with monitoring platforms or used independently to analyze customer feedback, reviews, and social media posts. Key features include:

<u>Sentiment classification:</u>

Categorization of text as positive, negative, or neutral.

<u>Aspect-based sentiment analysis:</u>

Identification of specific aspects or attributes mentioned in the text and their associated sentiment.

<u>Multi-language support:</u>

Ability to analyze text in multiple languages for global monitoring.

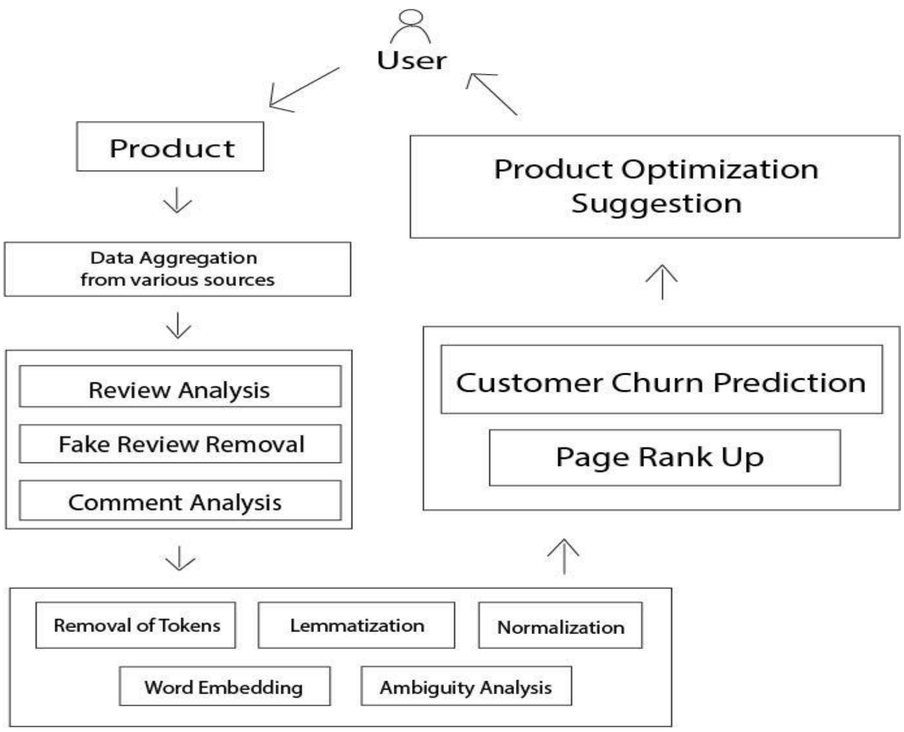
<u>Customizable sentiment lexicons:</u>

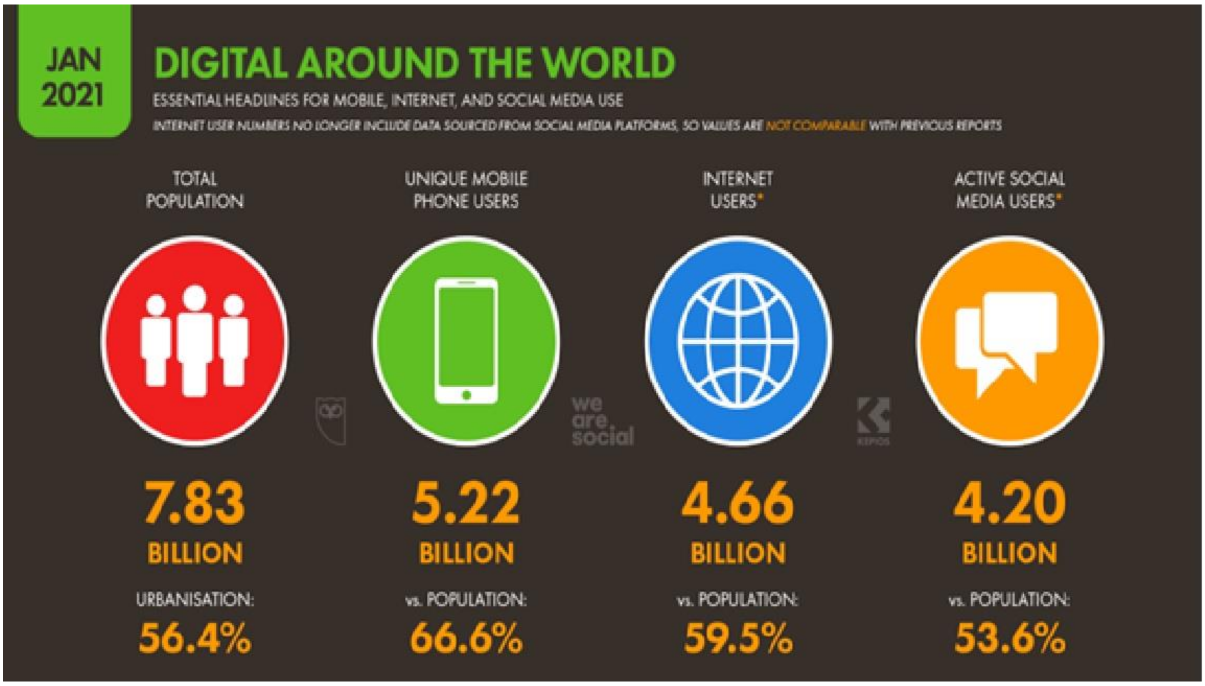
Tailoring sentiment analysis models to specific industries or domains.

Prominent sentiment analysis software solutions include:

- Lexalytics
- MonkeyLearn
- IBM Watson Natural Language Understanding
- Clarabridge
- Aylien

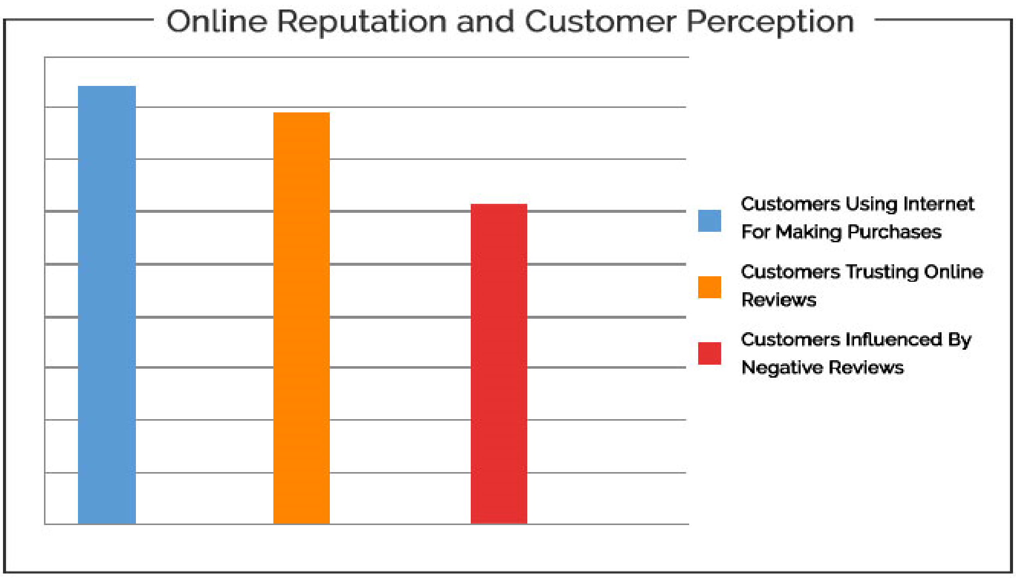
4.2 ARCHITECTURE DIAGRAM

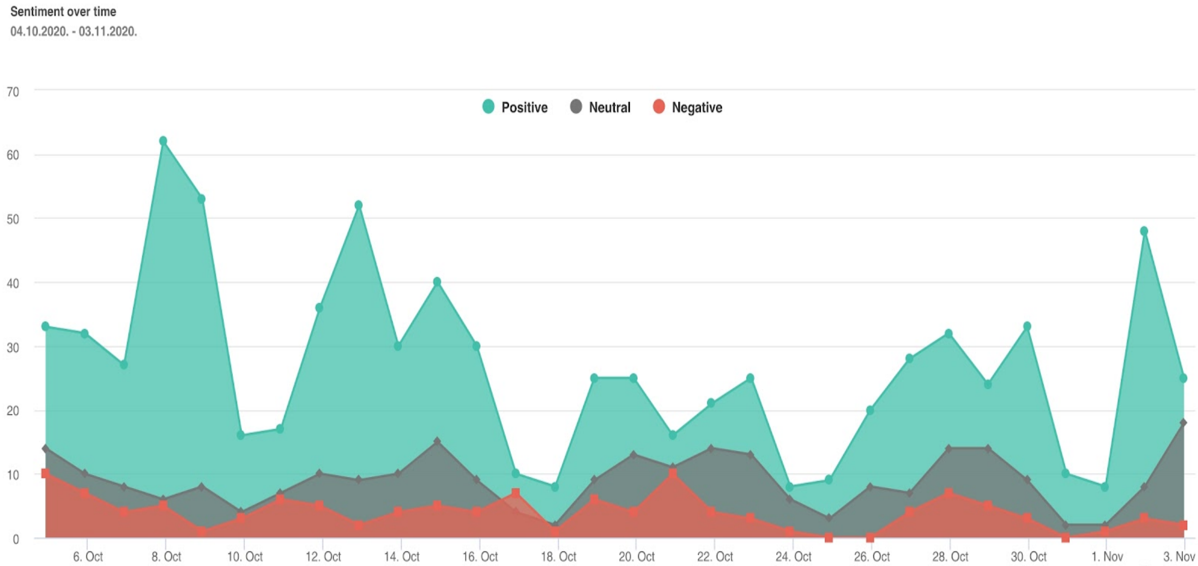




4.3 Data Visualization

The findings were visually presented using charts, graphs, and other visual aids to enhance the presentation of key insights and trends, making the data more accessible to the reader.





4.3.1 Ethical Considerations

Throughout the research process, ethical considerations were upheld. Informed consent was obtained from participants in interviews and surveys, and steps were taken to ensure the confidentiality and anonymity of respondents. Additionally, all data sources, including secondary data from literature and online platforms, were appropriately cited and referenced.

4.3.2 Limitations

It is important to acknowledge the limitations of this study, including potential bias in participant responses, the availability of public data for analysis, and the generalizability of findings from case studies.

CHAPTER 5

CONCLUSION

This project facilitates data-handling for large scale business networks. Users can integrate the features with their own software and real-time databases, and hence incorporating innovative solutions for their business needs. Maintenance is made possible for user-specific requests and queries. Natural Language Processing enables the users to process and analyze the data from the reviews provided by the users.

Users will mostly encounter two different types of people in a business environment namely Customers and Dealers.

Customers are the type of people who buy various products from the users. They review the product and rate them accordingly.

Dealers are the type of people who sell the raw materials to the users required to create a sellable product. These are divided into two different databases resulting in a well-structured and orderly data handling.

In order to run a successful business, we need to be up-to-date with the latest changes in the market and perform competitively. Features like Stock value prediction and Pagerank help the users to compare their business to other businesses and know their standing in the market.

Pagerank algorithm has been modified to tackle its demerits and provide better and ambiguous results. Finally, all the different features have been integrated with a user-friendly interface.

FUTURE WORK

This chapter focuses on providing comprehensive recommendations based on the findings and insights derived from the preceding chapters. It is divided into three main sections: best practices for product monitoring, strategies for effective reputation management, and suggestions for improvement.

Discuss the importance of quality control measures. Offer specific recommendations for setting up quality control protocols. Provide examples of quality control frameworks used in successful organizations. Emphasize the role of technology in data-driven product monitoring. Recommend specific tools and software for monitoring product performance. Provide case studies illustrating the successful implementation of technology in product monitoring. Discuss the significance of KPIs in product monitoring. Recommend a set of KPIs tailored to different industries. Provide guidance on how to track and measure KPIs effectively. Stress the value of customer feedback in product improvement. Suggest strategies for collecting and analyzing customer feedback. Highlight the importance of proactive reputation management. Recommend strategies for building and reinforcing a positive corporate image. Offer guidance on developing corporate social responsibility (CSR) initiatives.

Discuss strategies for crisis prevention and preparedness. Provide a framework for effective crisis management. Share case studies of organizations that successfully navigated reputation crises. Emphasize the significance of online reputation. Recommend approaches to monitor and engage with online audiences. Share examples of companies that have effectively managed their online reputation.

Advocate for a culture of continuous learning and adaptation. Suggest strategies for organizations to stay current with evolving trends. Share examples of companies that have successfully adapted to changing landscapes. Recommend methods for measuring the impact of implemented recommendations. Provide guidance on data collection and analysis to evaluate success. Offer a framework for ongoing improvement

CHAPTER 6
REFERENCES

1. “Online Reputation Management in the era of social media: A Review and Research Agenda”, Gianluigi Guido, Daniel Dalli, International Journal of Electronic Commerce, 2023.
2. “The Impact of Online Reputation on Consumer Behaviour in the E-Commerce Age: A Review and Meta-Analysis”, Kholekile Gwebu, Jingguo Wang, Journal of Retailing, 2023.
3. “Product Monitoring and Reputation Management in the Age of Big Data: A Review and Research Directions”, Gohar F. Khan, Xueming Luo Frontiers in Computer Science, 2023.
4. “The Role of Artificial Intelligence in Online Reputation Management: A Review”, Yubo Chen, Huaxia Rui Journal of Public Relations Research, 2023.
5. “A Blockchain-Based Framework for Product Monitoring and Reputation Management”, Qian Li, Liangmin Wang, Xiaojun Chen, IEEE Access, 2023.
6. “Analysing and detecting review spam”, Bing Liu, department of computer science university of Illinois at Chicago, 2008.
7. “Fake product review monitoring & removal and sentiment analysis of genuine reviews”, Abishek Pund, Ramteke Sanchit, Shinde Shailesh, International Journal of Engineering and Management Research, 2019.
8. “Fake Product Review Monitoring and Removal using Opinion Mining”, Gyandeep Dowari, Dibya Jyoti Bora, IEEE conference publication, 2020.
9. “Social Media Monitoring and Reputation Management for Businesses: A Review of the Literature and Future Research Directions”, Mayank Yadav, Zillur Rahman, International Journal of Business and Management, 2022.
10. “The Impact of Online Reputation on Consumer Purchase Decisions: A Review of the Literature”, Mirella Yani-de-Soraino, Chris Rowley Frontiers in Psychology, 2022.
11. “A Machine Learning Approach to Fake Product Review Detection”, Jungdong Li, Jianming Hu, Hongzhi Ying, Cheng Yang, IEEE Transactions on Knowledge and Data Engineering, 2022.
12. “The Role of Artificial Intelligence in Online Reputation Management: A Review”, Elad Segev, Yaniv Gvill, Journal of Public Relations Research, 2022.

Program Code and Output

CUSTOMER RELATIONSHIP MANAGEMENT(CRM) CODE:

```
import streamlit as st

from streamlit_option_menu import option_menu

from urllib3 import disable_warnings

from urllib3.exceptions import InsecureRequestWarning

disable_warnings(InsecureRequestWarning)

import pyrebase

from firebase_admin import db

firebaseConfig = {

    'apiKey': " AIzaSyCwdhswGJY5QMANM4bVO8XQmd1TJ08pM7Y ",

    'authDomain': "app001-97f05.firebaseapp.com",

    'projectId': "app001-97f05",

    'storageBucket': "app001-97f05.appspot.com",

    'messagingSenderId': "461483901137",

    'appId': "1:461483901137:web:df034f8dccc390a20c45f0",

    'measurementId': "G-RE5B8WK68Z",

    'databaseURL' : 'https://app002-f6090-default-rtdb.firebaseio.com/'

}

firebase = pyrebase.initialize_app(firebaseConfig)

auth = firebase.auth()

db = firebase.database()
```

```
def signup(email,password):
    try:
        auth.create_user_with_email_and_password(email,password)

        st.toast("Registered New User Successfully")

        return 1

    except:

        login(e,p)

        return 0


def login(e,p):

    try:

        auth.sign_in_with_email_and_password(e,p)

        st.toast("Logged in Successfully")

    except:

        st.error('Invalid Credentials')



if 'clicked' not in st.session_state:

    st.session_state['clicked'] = False

if 'clicked' not in st.session_state:

    st.session_state.clicked = False

def click_button():

    st.session_state.clicked = True




st.header('Real-time Data Dashboard')

selected = option_menu(

    menu_title = None,
```

```
    options = ["Data Entry","Data Overview","Product DB"],
    icons = ["pencil-fill","database-fill-check","database-lock"],
    orientation = 'horizontal',
)
if selected == 'Data Overview':
    st.write("Toggle to switch between Cutomers and Dealers ")
    x = st.toggle("Toggle Button")
    if x:
        users = db.child("users").order_by_child("Type").equal_to("Dealer").get()
        st.write(users.val())
    else:
        users = db.child("users").order_by_child("Type").equal_to("Customer").get()
        st.write(users.val())

if selected == 'Data Entry':
    with st.form("Enter Details here:",clear_on_submit = True):
        d = st.selectbox(label='Are you a Customer or a
Dealer',options=['Customer','Dealer'])
        a = st.text_input(label='Enter Name')
        e = st.text_input(label='Enter Email')
        p = st.text_input(label='Enter Password')
        submit = st.form_submit_button(label = "Submit")
        if submit:
            if a != '':
                data = { "Name" : a, "Type" : d, "Email" : e, "Password" : p }
```

```
            y = signup(e,p)

            if y == 1:

                db.child("users").child(a).set(data)

                st.toast("Submitted Details Successfully!")


if selected == 'Product DB':

    select = st.selectbox(label = '', options = ["Enter Product Details","Product
Overview"])

    if select == 'Enter Product Details':

        with st.form("Product Details",clear_on_submit = True):

            prodName = st.text_input("Enter the name of the Product")

            prodQuantity = st.number_input("Enter Product
quantity",min_value=1,max_value=100,step=1)

            submit = st.form_submit_button(label = "Submit")

            if submit:

                d = {"Name" : prodName, "Quantity" : prodQuantity}

                if prodName != '':

                    products =
db.child("Products").order_by_child("Name").equal_to(prodName).get()

                    if products == '' :

                        db.child("Products").child(prodName).set(d)

                    else:

                        q = db.child("Products").child(prodName).get()

                        q1 = q.val()

                        q2 = q1['Quantity']

                        q2 -= prodQuantity

                        d = {"Name" : prodName, "Quantity" : q2}
```

```
                db.child("Products").child(prodName).set(d)
            st.toast("Product Details Saved!")
    if select == 'Product Overview':
        products = db.child("Products").order_by_child("Name").get()
        st.write(products.val())
```

PAGE RANK CODE:

```
import numpy as np
import streamlit as st

def pagerank(nodes, edges, damping_factor=0.85, max_iterations=100,
tolerance=1e-6):
    num_nodes = len(nodes)
    adjacency_matrix = np.zeros((num_nodes, num_nodes))
    incoming_count = np.zeros(num_nodes)

    for node, incoming_nodes in edges.items():
        node_index = nodes.index(node)
        for incoming_node in incoming_nodes:
            incoming_index = nodes.index(incoming_node)
            adjacency_matrix[node_index, incoming_index] = 1
            incoming_count[incoming_index] += 1

    for j in range(num_nodes):
        if incoming_count[j] != 0:
            adjacency_matrix[:, j] /= incoming_count[j]
```

```
    pagerank_scores = np.ones(num_nodes) / num_nodes

    for _ in range(max_iterations):

        new_pagerank_scores = (1 - damping_factor) / num_nodes + damping_factor *
np.dot(adjacency_matrix, pagerank_scores)

        if np.linalg.norm(new_pagerank_scores - pagerank_scores, 1) < tolerance:

            break

        pagerank_scores = new_pagerank_scores

    return pagerank_scores


st.title("PageRank Calculator")

select = st.selectbox(label = "Select Type of Pagerank", options = ["Default",
"Modified"])

num_nodes = st.number_input("Enter number of nodes:", min_value=1, step=1)

nodes = st.text_input("Enter Name of nodes (separated by space):")

nodes = nodes.split() if nodes else []

if len(nodes) != num_nodes:

    st.warning("Please provide correct number of node names.")


edges = {}

for node in nodes:

    edges[node] = st.text_input(f"Enter incoming nodes to node {node} (separated by
space):")

    edges[node] = edges[node].split() if edges[node] else []


def pagerankM(nodes, edges, damping_factor=0.85, relevance_factor=0.1,
max_iterations=100, tolerance=1e-6):

    num_nodes = len(nodes)

    adjacency_matrix = np.zeros((num_nodes, num_nodes))
```

```
    incoming_count = np.zeros(num_nodes)
    for node, incoming_nodes in edges.items():
        node_index = nodes.index(node)
        for incoming_node in incoming_nodes:
            incoming_index = nodes.index(incoming_node)
            adjacency_matrix[node_index, incoming_index] = 1
            incoming_count[incoming_index] += 1

    for j in range(num_nodes):
        if incoming_count[j] != 0:
            adjacency_matrix[:, j] /= incoming_count[j]

    pagerank_scores = np.ones(num_nodes) / num_nodes
    for _ in range(max_iterations):
        new_pagerank_scores = (1 - damping_factor) / num_nodes + damping_factor * 
np.dot(adjacency_matrix, pagerank_scores)
        new_pagerank_scores += relevance_factor * np.ones(num_nodes) / 
num_nodes
        if np.linalg.norm(new_pagerank_scores - pagerank_scores, 1) < tolerance:
            break
        pagerank_scores = new_pagerank_scores
    return pagerank_scores


if st.button("Calculate PageRank"):
    st.write("PageRank Scores")
    if select == 'Modified':
        scores = pagerankM(nodes, edges)
```

```
        sorted_indices = np.argsort(scores)[::-1]
        for i in sorted_indices:
            st.write(f"{nodes[i]}: {(scores[i]*100):.2f}")

    elif select == 'Default':
        scores = pagerank(nodes, edges)
        sorted_indices = np.argsort(scores)[::-1]
        for i in sorted_indices:
            st.write(f"{nodes[i]}: {(scores[i]*100):.2f}")
```

REVIEW ANALYSIS CODE:

```
from textblob import TextBlob
import pandas as pd
import streamlit as st
import cleantext
from PIL import Image

st.title('Product Management')
with st.expander('Analyze Text'):
    text = st.text_input('Enter Text:')
    if text:
        blob = TextBlob(text)
        st.write('Polarity: ',round(blob.sentiment.polarity),2)
        st.write('Subjectivity: ',round(blob.sentiment.subjectivity),2)
with st.expander('Process Text '):
```

```
    pre = st.text_input('Clean Text:')

    if pre:

        st.write(cleantext.clean(pre,clean_all=False,extra_spaces=True,stopwords=Tru
e,lowercase=True,numbers=True,punct=True))

with st.expander('Dataset Analysis'):

    upl = st.file_uploader('Upload File')


    if st.checkbox("Preview:"):

        df = pd.read_csv(upl,encoding='utf-8',encoding_errors='ignore')

        df['Review'] = df['selected_text']

        df['Analysis'] = df['sentiment']

        st.dataframe(df[['Review','Analysis']])

        @st.cache_data
        def convert_df(df):

            return df.to_csv().encode('utf-8')

        csv = convert_df(df)

        st.download_button(
            label="Download Data as CSV",

            data=csv,

            file_name = 'sentiment.csv',

            mime='text/csv',
        )

st.write('\n\n\n\n\n')

st.title('Product Analysis from Image Input')

uploaded_file = st.file_uploader('Choose Image to upload…', type = (["jpg", "jpeg"]))

if uploaded_file is not None:
```

```
    img = Image.open(uploaded_file)
    st.image(img, caption = 'Uploaded image')
```

STOCK CODE:

```
import streamlit as st
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

st.title('Stock Market Prediction')
uploaded_file = st.file_uploader("Upload CSV file", type=['csv'])

if uploaded_file is not None:
    data = pd.read_csv(uploaded_file)
    st.subheader('Raw data')
    st.write(data)
    st.subheader('Data preprocessing')

    data['Date'] = pd.to_datetime(data['Date'])
    data['Year'] = data['Date'].dt.year
    data['Month'] = data['Date'].dt.month
    data['Day'] = data['Date'].dt.day


    X = data[['Year', 'Month', 'Day']]
    y = data['Close']
```

```
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

    st.subheader('Model training')
    model = LinearRegression()
    model.fit(X_train, y_train)

    st.subheader('Model evaluation')
    y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    st.write('Mean Squared Error:', mse)
    st.write('R^2 Score:', r2)

    st.subheader('Make a prediction')

    year = st.slider('Year', min_value=2010, max_value=2030, value=2022)
    month = st.slider('Month', min_value=1, max_value=12, value=1)
    day = st.slider('Day', min_value=1, max_value=31, value=1)
    prediction = model.predict([[year, month, day]])
    st.write('Predicted Closing Price:', prediction[0])
```

WORKSHOP CODE:

```
import streamlit as st
from streamlit_option_menu import option_menu
from urllib3 import disable_warnings
from urllib3.exceptions import InsecureRequestWarning
disable_warnings(InsecureRequestWarning)
```

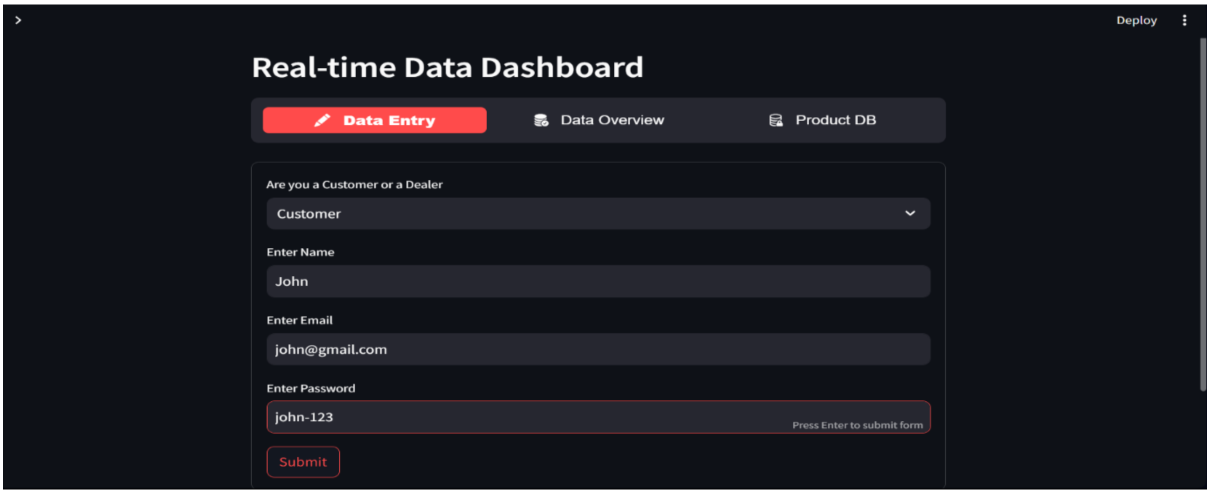
```
import pyrebase
from firebase_admin import db
firebaseConfig = {
    'apiKey': " AIzaSyCwdhswGJY5QMANM4bVO8XQmd1TJ08pM7Y ",
    'authDomain': "app001-97f05.firebaseapp.com",
    'projectId': "app001-97f05",
    'storageBucket': "app001-97f05.appspot.com",
    'messagingSenderId': "461483901137",
    'appId': "1:461483901137:web:df034f8dccc390a20c45f0",
    'measurementId': "G-RE5B8WK68Z",
    'databaseURL' : 'https://app002-f6090-default-rtdb.firebaseio.com/'
}
firebase = pyrebase.initialize_app(firebaseConfig)
auth = firebase.auth()
db = firebase.database()

st.header('Workshop')
selected = option_menu(
    menu_title = None,
    options = ["Materials","Produce"],
    icons = ["list-check","database","hammer"],
    orientation = 'horizontal',
)

if selected == "Materials":
```

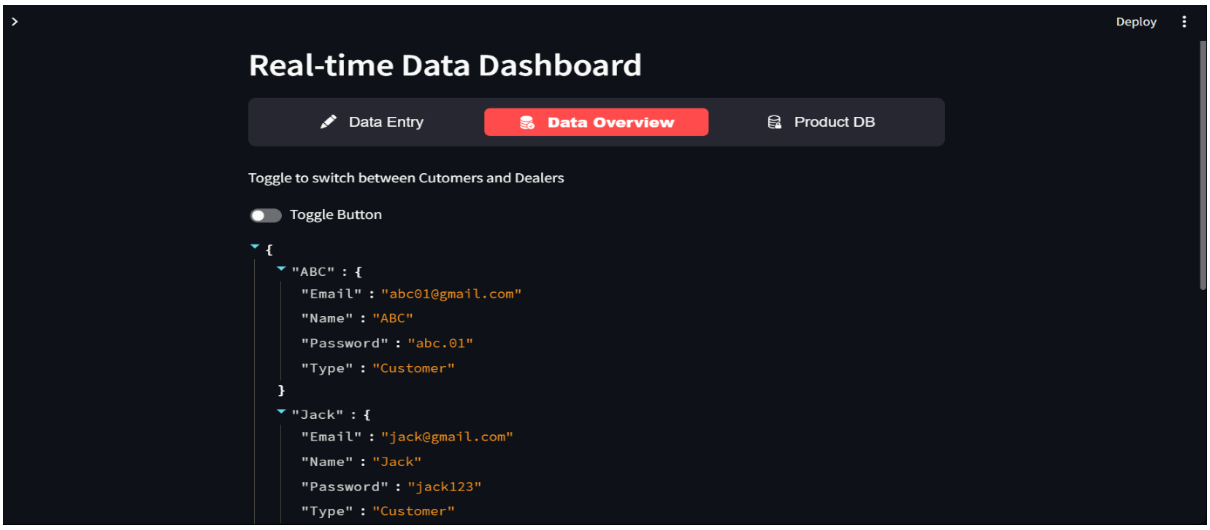
```
    toggle = st.toggle(label = "View Available Materials")

    if toggle:

        items = db.child("Materials").order_by_child("Name").get()

        st.write(items.val())

    else:

        with st.form("Order Materials ",clear_on_submit = True):

            Name = st.text_input("Enter the name of the material")

            Quantity = st.number_input("Enter
quantity",min_value=1,max_value=100,step=1)

            submit = st.form_submit_button(label = "Submit")

            if submit:

                d = {"Name" : Name, "Quantity" : Quantity}

                if Name != '':

                    products =
db.child("Materials").order_by_child("Name").equal_to(Name).get()

                    if products == '' or products == None:

                        db.child("Materials").child(Name).set(d)

                    else:

                        q = db.child("Materials").child(Name).get()

                        q1 = q.val()

                        if q1 == None:

                            db.child("Materials").child(Name).set(d)

                        else:

                            q2 = q1['Quantity']

                            q2 += Quantity
```

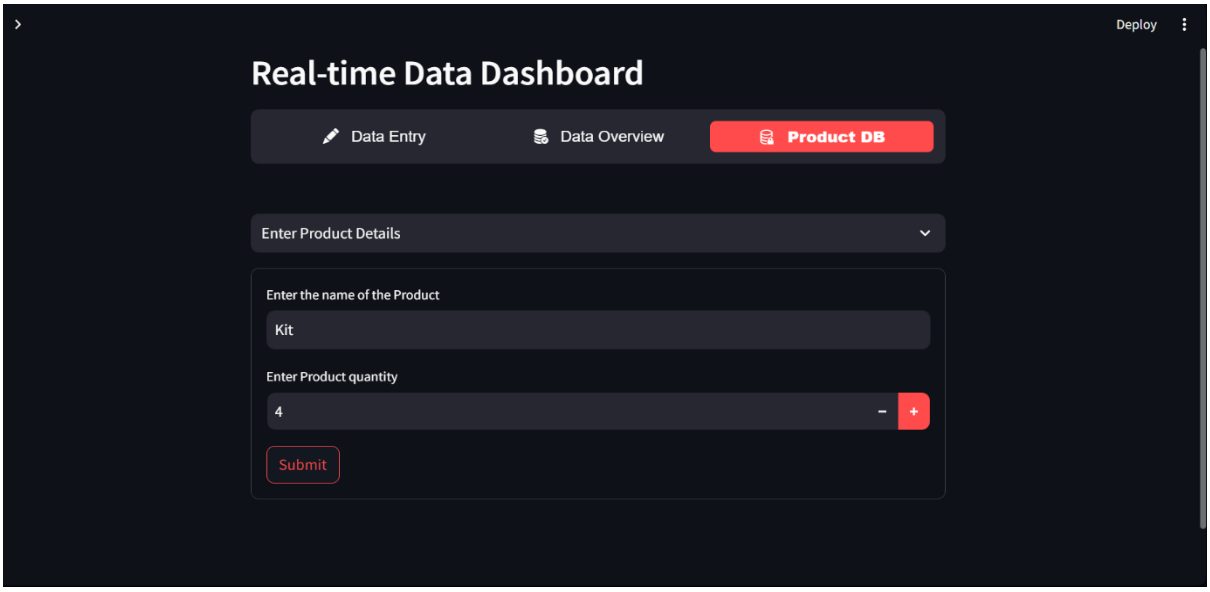
```
                d = {"Name" : Name, "Quantity" : q2}
                db.child("Materials").child(Name).set(d)
            st.toast("Successfully Ordered!")
        else:
            st.warning("Name cannot be empty")
if 'clicked' not in st.session_state:
    st.session_state['clicked'] = False
if 'clicked' not in st.session_state:
    st.session_state.clicked = False
def click_button():
    st.session_state.clicked = True

if selected == "Produce":
    products = db.child("Products").order_by_child("Name").get()
    option = list(products.val())
    with st.form("Select Item to Produce", clear_on_submit = True):
        item = st.selectbox(label = "Select an item to produce", options = option)
        if item == 'Kit':
            pen1 = db.child("Materials").child("Pen").child("Quantity").get()
            pen = pen1.val()
            pencil1 = db.child("Materials").child("Pencil").child("Quantity").get()
            pencil = pencil1.val()
            scale1 = db.child("Materials").child("Scale").child("Quantity").get()
            scale = scale1.val()
            maxi = max(pen,pencil,scale)
```

```
            amount = st.number_input("Enter Number of Items to Produce",min_value = 
1,max_value = maxi, step = 1)

        submitted = st.form_submit_button("Submit",on_click = click_button())

        if submitted:

            if pen < amount or pencil < amount or scale < amount:

                st.error("Insufficient Materials. Order more to produce.")


            else:

                pen -= amount

                pencil -= amount

                scale -= amount

                data = {"Name" : "Pen", "Quantity" : pen}

                db.child("Materials").child("Pen").set(data)

                data = {"Name" : "Pencil", "Quantity" : pencil}

                db.child("Materials").child("Pencil").set(data)

                data = {"Name" : "Scale", "Quantity" : scale}

                db.child("Materials").child("Scale").set(data)

                st.toast("Order Success")

        itemQuantity = db.child("Products").child(item).child("Quantity").get()

        itemq = itemQuantity.val()

        q = itemq + amount

        g = {"Name" : item, "Quantity" : q}

        db.child("Products").child(item).set(g)
```

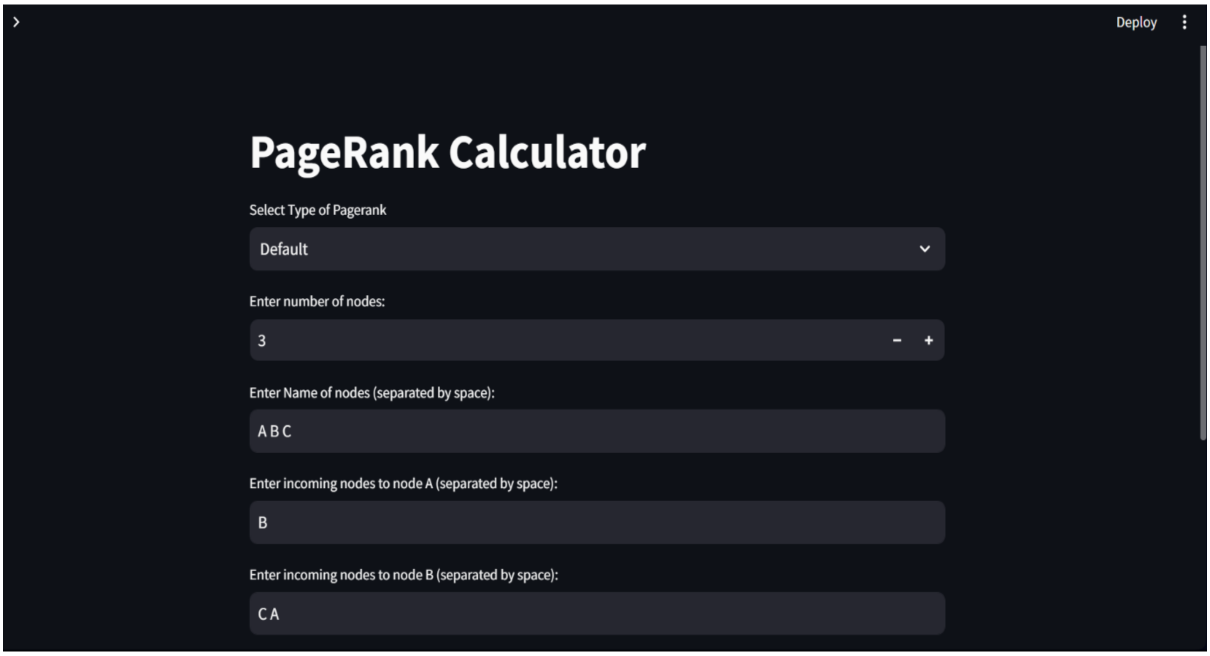
CUSTOMER RELATIONSHIP MANAGEMENT(CRM) OUTPUT:

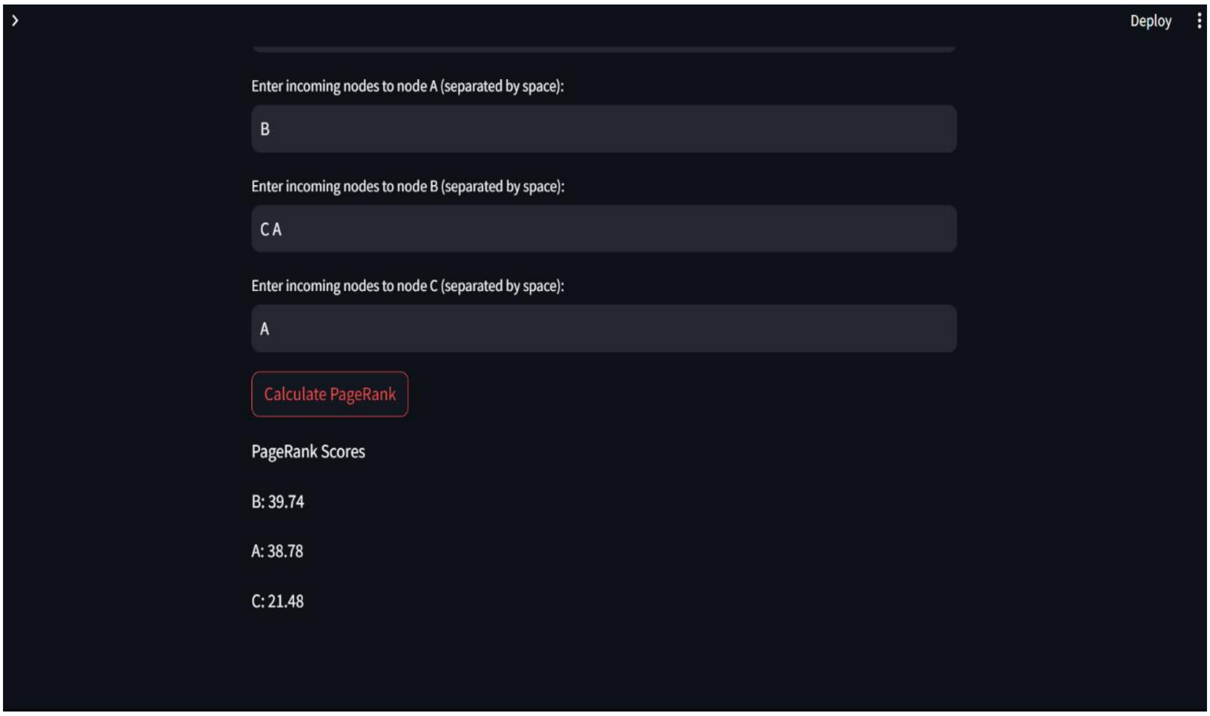




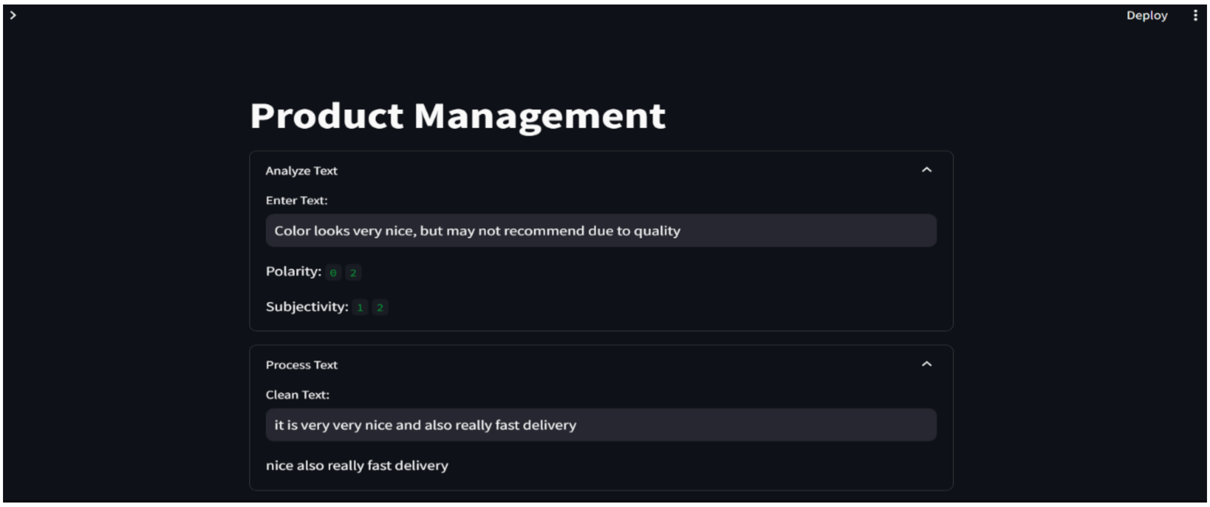


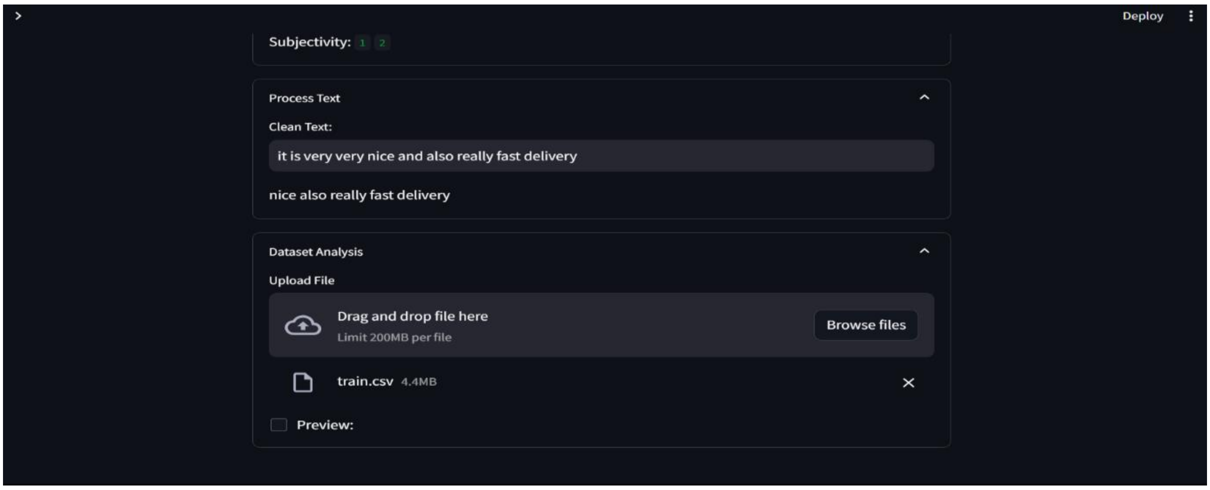
PAGE RANK OUTPUT:

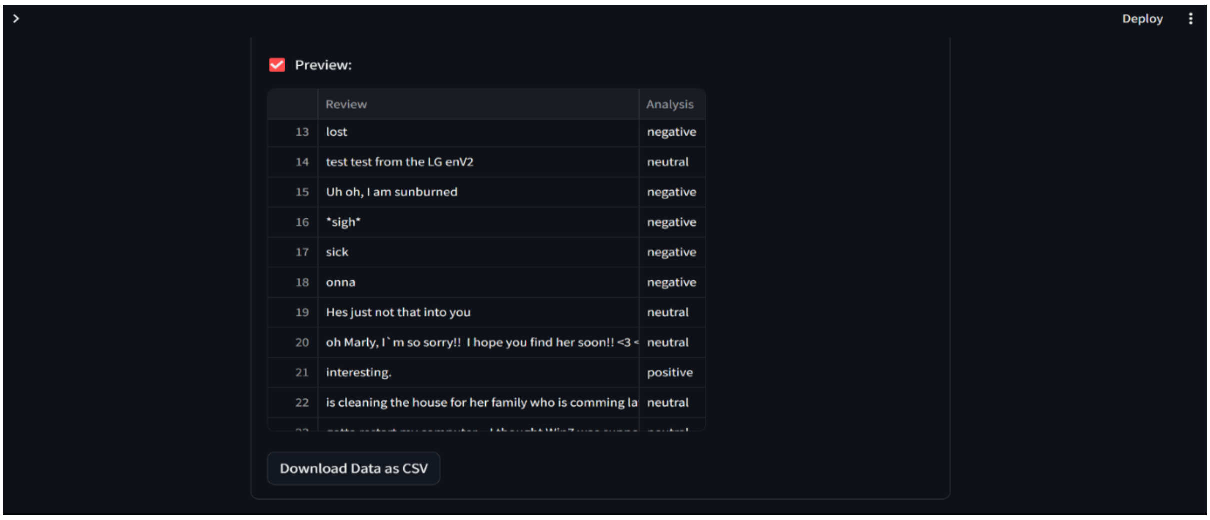


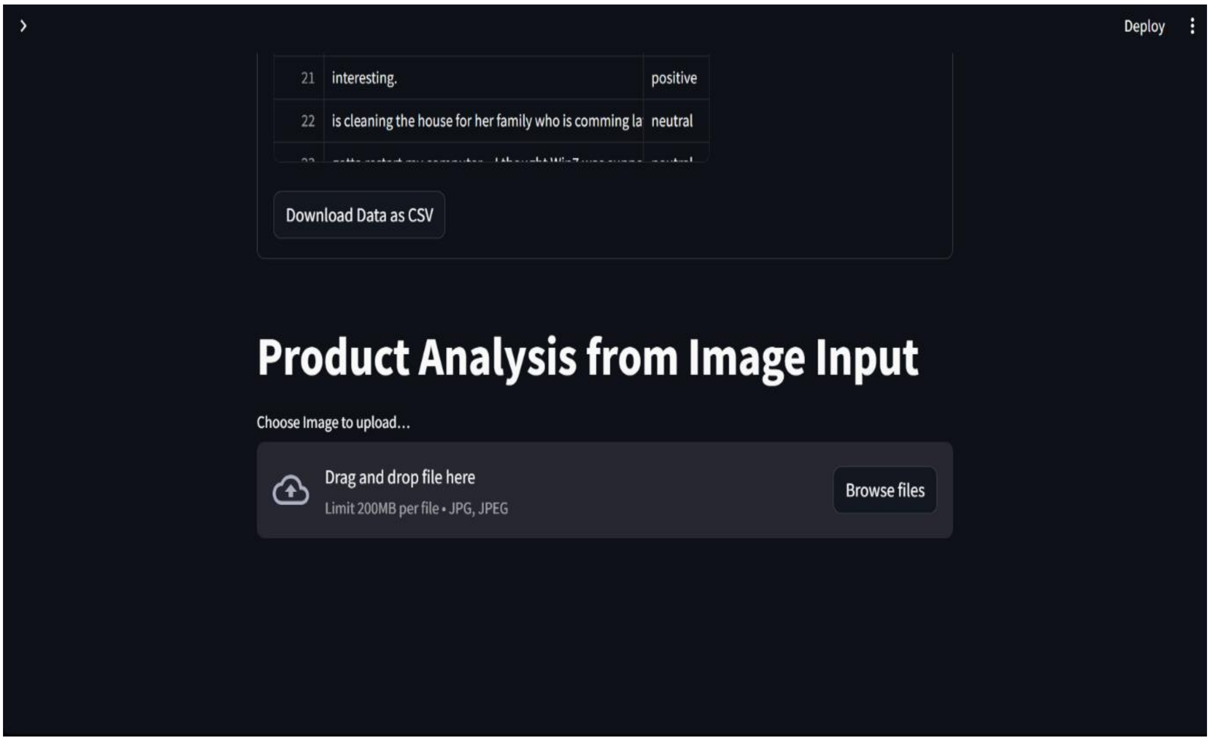


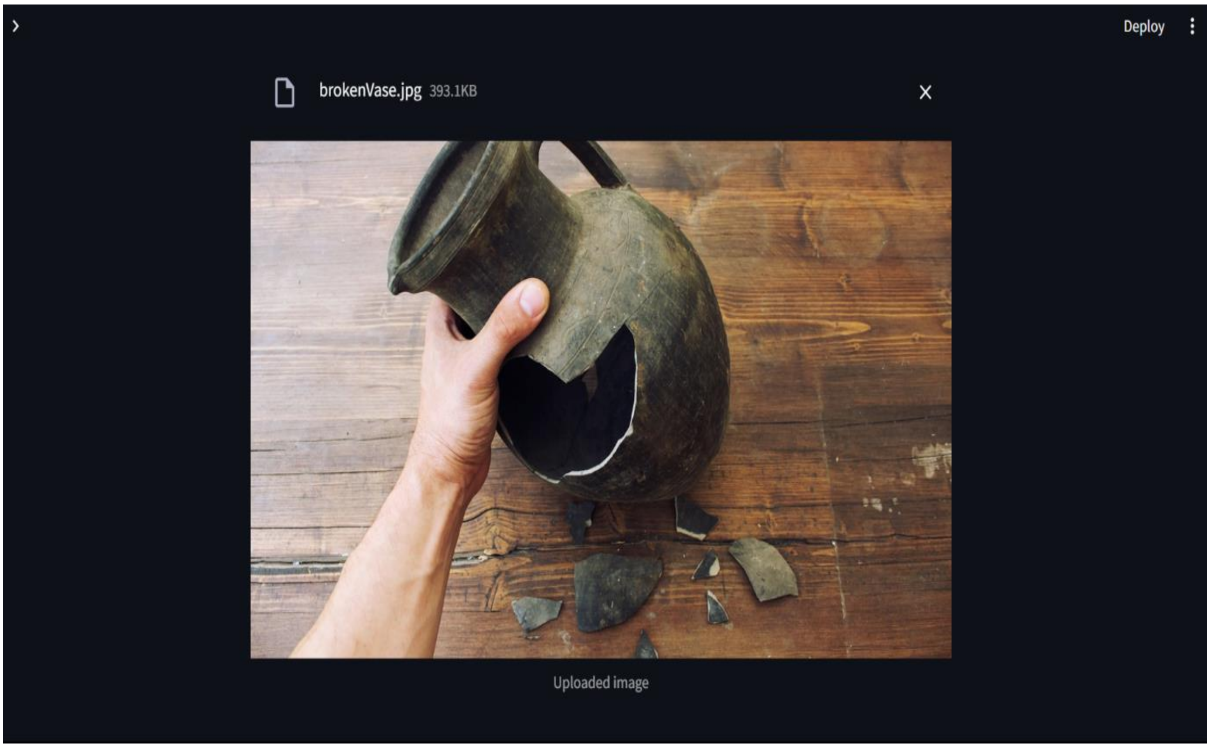
REVIEW ANALYSIS OUTPUT:



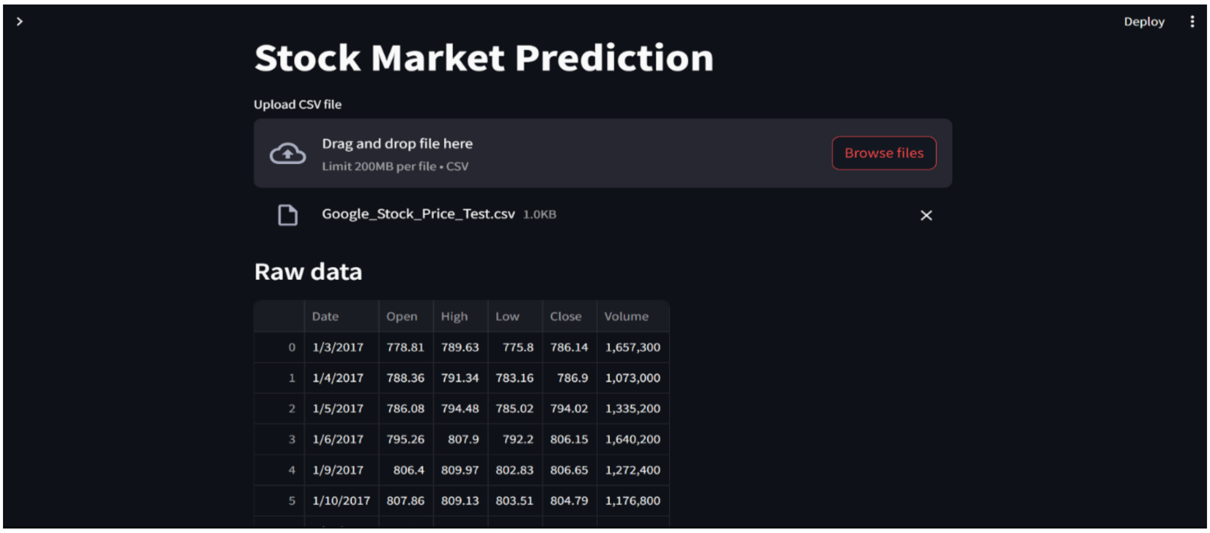


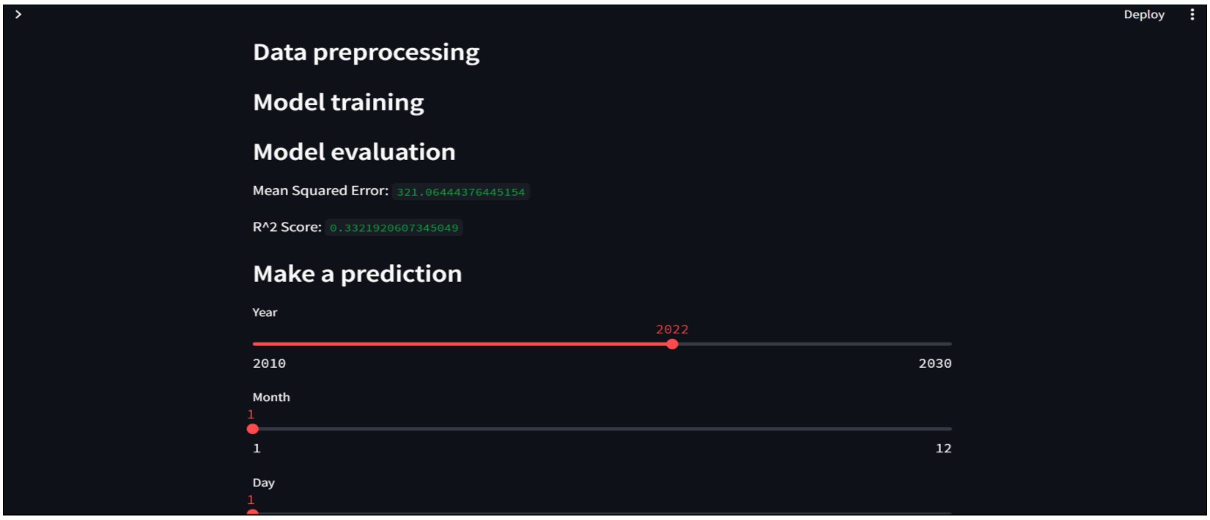



Deploy
21 interesting. positive
22 is cleaning the house for her family who is comming la neutral
Download Data as CSV
Product Analysis from Image Input
Choose Image to upload...
Drag and drop file here
Limit 200MB per file • JPG, JPEG
Browse files


Deploy
brokenVase.jpg 393.1KB
Uploaded image

STOCK OUTPUT:





WORKSHOP OUTPUT: