PHASE-3

**STUDENT NAME: Jasmine Joicy. A**

**REGISTER NUMBER: 422223243025**

**INSTITUTION: Surya Group Of Institutions**

**DEPARTMENT: B.Tech Artificial Intelligence and Data Science**

**DATE OF SUBMISSION:**

**GITHUB REPOSITORY LINK:**

**PROBLEM STATEMENT:**

The topic is **“Revolutionizing customer support with an intelligent chatbox for automated assistance”.** This topic is based on a real world problem. Many products has customer service,if the customer has a query or a problem with the product the customer can directly contact the customer service through calls,emails etc…The person working in the product has to reply to the customer and take actions according to it. This project recieves the customer reviews and produce automated assistance based on the customer feedback through an intelligent chatbox. This problem is based on regression.

**ABSTRACT:**

The dataset related to the problem is collected and analysed. Based on the data the user can identify what changes can be made to the product and improvement can be done relevantly. The chatbot itself analyses the data and gives the owner of the product an idea of what the customer is thinking of the project. The dataset is a collection of feedback of the product. For example it contains data like whether the customer is satisfied or not. The project aims to make an intellectual chatbot which make relevant changes by itself by observing the customer feedback. This project enables a strong relationship between the product seller and the product buyer. The outcome of the project is that the product seller achieve their marketing rate high by enabling this intellectual chatbot.

**SYSTEM REQUIREMENT:**

**HARDWARE:** An average RAM storage is needed to store the dataset.

**SOFTWARE:** Libraries such as python, colab, jupyter notebook is required.

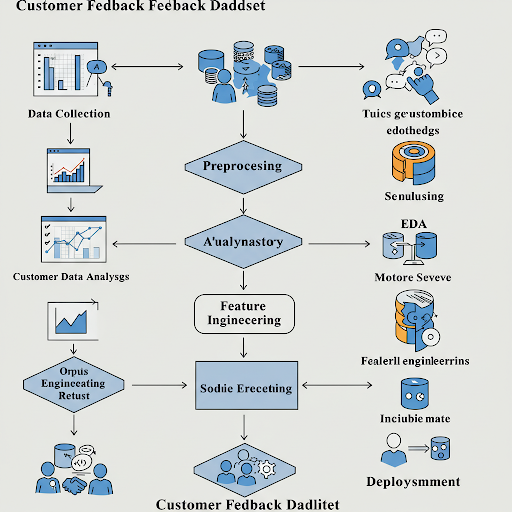
**OBJECTIVES:**

This project goal is to automatically save the response from the customer and automatically gives response to it to satisfy the customer

Key Technical objectives:

It’s main purpose is to reveal the customer’s level of satisfaction and help product, customer success, and marketing teams understand where there is room for improvement. The model aims to achieve the accuracy and real-world-applicability. The goal has changed after exploration not only giving response to the feedback it also helps us to find ways to improve the product. It also helps us to predict the outcome of the product after the changes made to the product according to the customer.

**FLOW CHART OF PROJECT WORKFLOW:**

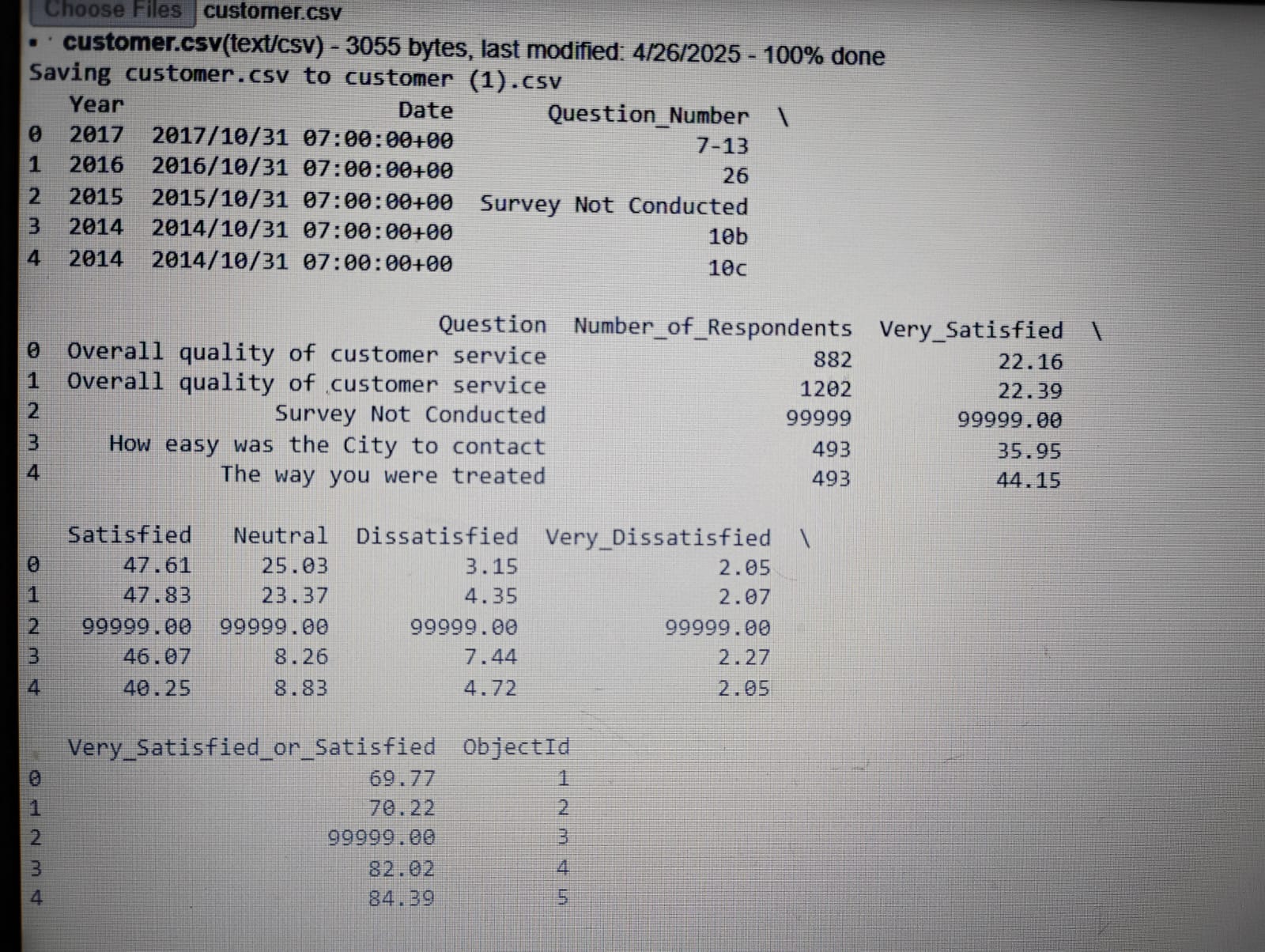
****

**DATASET DESCRIPTION:**

SOURCE: gts.ai

TYPE: Public

SIZE AND STRUCTURE: 29 rows and 12 columns

Df.head():

**DATA PREPROCESSING:**

**Handling missing values:** Verified the dataset there is no missing values.

**Duplicate records:** The dataset contains duplicate data. It is irrelevant to the data. Since, the duplicate data is an dependent data therefore it can also be removed for the purpose of the project.

duplicates = df.duplicated()

**Outliers:** Checked for the absence of outliers.

**Encoding categorical variables:** Label encoding is done using:

df['Product\_Encoded'] = label\_encoder.fit\_transform(df['Product'])

One hot encoding is done using:

product\_encoded = onehot\_encoder.fit\_transform(df[['Product']]) product\_df = pd.DataFrame(product\_encoded, columns=onehot\_encoder.get\_feature\_names\_out(['Product']))

**Standardization:** Numerical features are used for standardization.

**Transformation steps used:**

**Lowercasing text :** df['Comment\_Lower'] = df['Comment'].str.lower()

**Removing punctuation:** df['Comment\_NoPunct'] = df['Comment'].str.replace(r'[^\w\s]+', '', regex=True)

**Tokenization :** df['Comment\_Tokens'] = df['Comment'].str.split()

**Label encoding:**from sklearn.preprocessing import LabelEncoder; le = LabelEncoder(); df['Product\_Encoded'] = le.fit\_transform(df['Product'])

**One-Hot encoding:** df = pd.get\_dummies(df, columns=['Rating'], prefix=['Rating'])

**Min-Max scaling:** from sklearn.preprocessing import MinMaxScaler; scaler = MinMaxScaler(); df['Word\_Count\_Scaled'] = scaler.fit\_transform(df[['Word\_Count']])

**Calculating word count:** df['Word\_Count'] = df['Comment'].apply(lambda x: len(str(x).split()))

**Standardizing a numerical column:** from sklearn.preprocessing import StandardScaler; scaler = StandardScaler(); df['Word\_Count\_Standard'] = scaler.fit\_transform(df[['Word\_Count']])

**Apllying TF-IDF to the comment text:** from sklearn.feature\_extraction.text import TfidfVectorizer; tfidf = TfidfVectorizer(); tfidf\_matrix = tfidf.fit\_transform(df['Comment'])



**EXPLORATORY DATA ANALYSIS(EDA):**

Visaul tools like histogram, lineplots are used.

1. **Calculate correlations:** Use statistical methods (Pearson or Spearman) on numerical/encoded data.
2. **Visualize with heatmaps:** See patterns of positive and negative relationships at a glance.
3. **Look for strong links:** Identify variables that consistently move together (positive) or opposite (negative).
4. **Note clusters:** Observe groups of highly correlated variables for underlying themes.
5. **Remember correlation isn't causation:** Further investigation is needed to understandwhy relationships exist.

**Key Takeaways and Insights:**

1. **General sentiment:** Understand if customers are mostly happy, neutral, or unhappy.
2. **Drivers of sentiment:** Identify specific factors boosting or hurting satisfaction.
3. **Problem areas:** Pinpoint recurring issues needing improvement.
4. **Strengths to leverage:** Recognize what's working well and can be expanded.



**FEATURE ENGINEERING:**

1. Textr-Based feature
2. Meta-based feature(is available)
3. Interaction feature(Combining existing features)

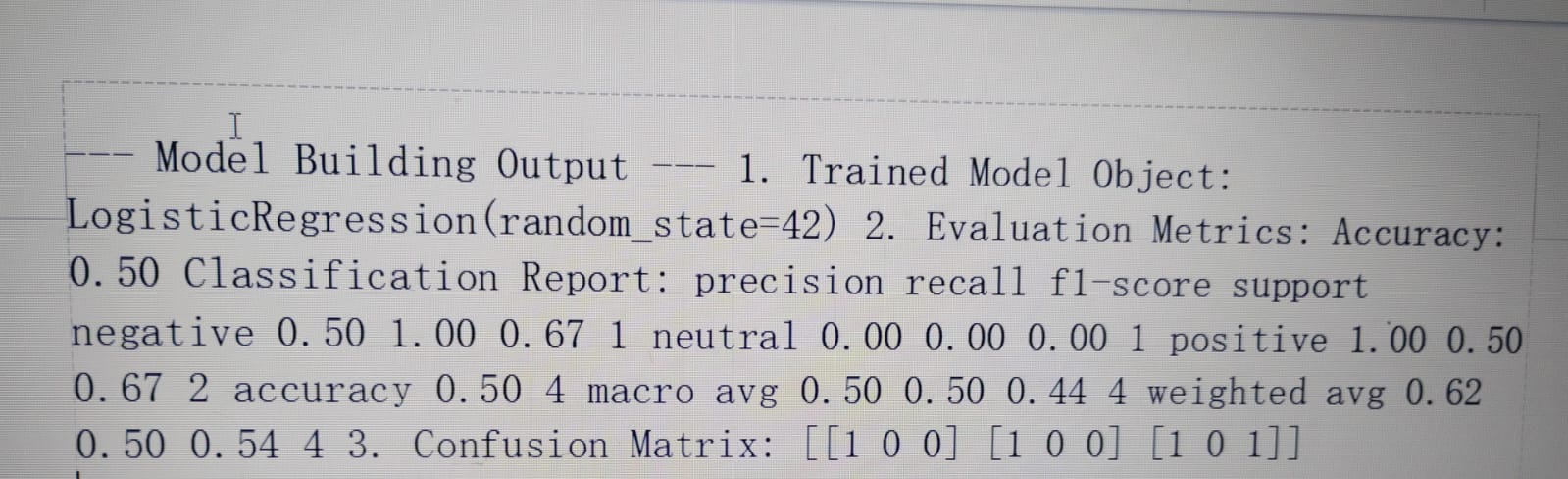
Combined and split techniques are also used as a part of feature engineering.

**MODEL BUILDING:**

Logistic regression and decision tree algorithm which are binary classification algorithm can be used here. These models are used here because it is a linear model that can work surprisingly well for text classification, especially with well-engineered features and to perform interpretable classification or regression tasks like sentiment analysis or satisfaction prediction.

Models can be trained based on its classification and regression.

**Output of the model building:**

****

MODEL EVALUATION:

 **Evaluation Metrics:**

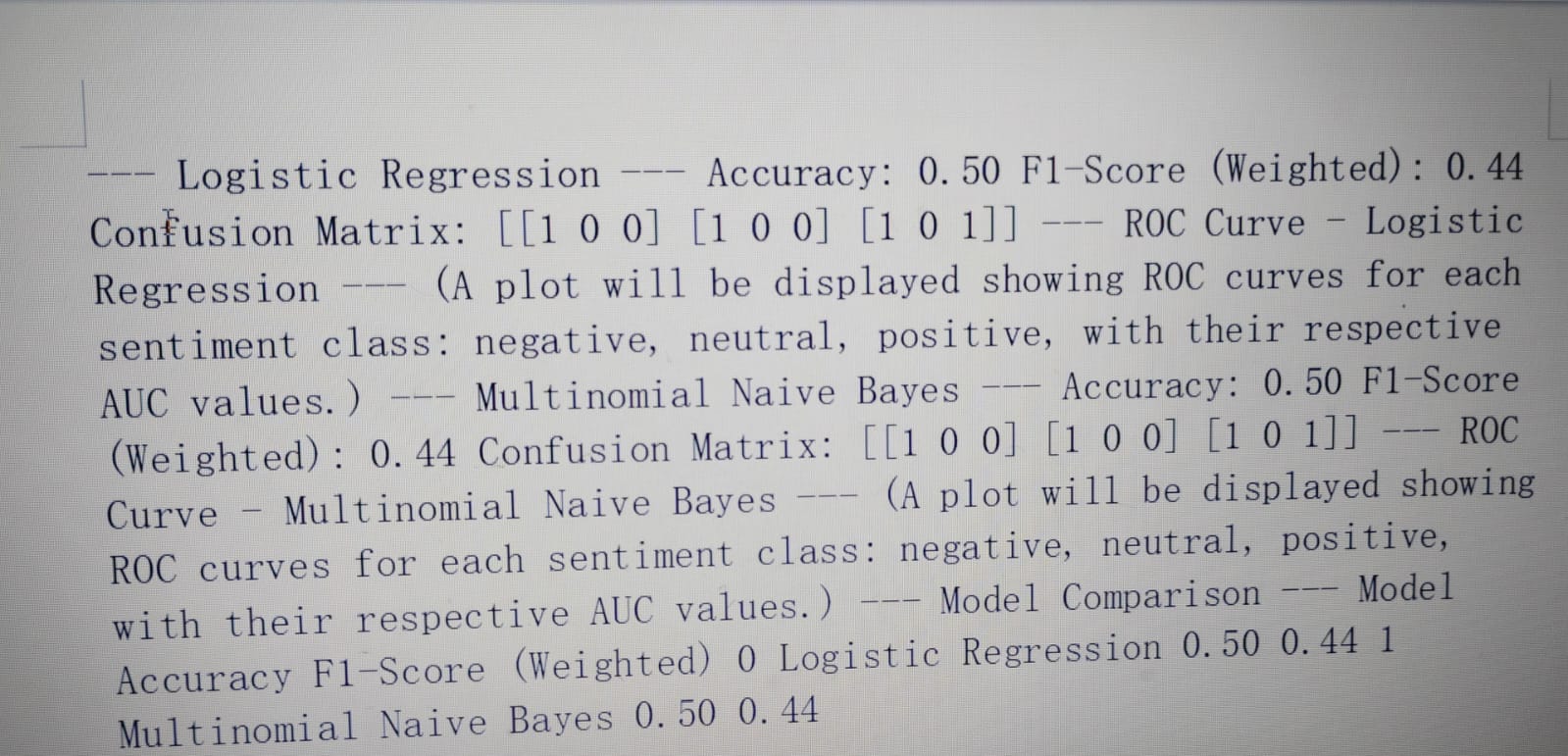
* **Accuracy:** The overall percentage of correctly classified customer feedback sentiments.
* **F1-Score (Weighted):** A balanced measure of precision and recall, considering the number of instances for each sentiment (negative, neutral, positive).

 **Visuals:**

* **Confusion Matrix:** A table showing how many feedback instances were correctly and incorrectly classified for each sentiment. The rows represent the actual sentiments, and the columns represent the predicted sentiments.
* **ROC Curve (Receiver Operating Characteristic):** A graph plotting the True Positive Rate (sensitivity) against the False Positive Rate (1 - specificity) for each sentiment class at various classification thresholds. The Area Under the Curve (AUC) indicates how well the model can distinguish between positive and negative sentiments for each class. You'll see a separate curve for "negative" vs. "not negative," "neutral" vs. "not neutral," and "positive" vs. "not positive."

 **Error Analysis / Model Comparison Table:**

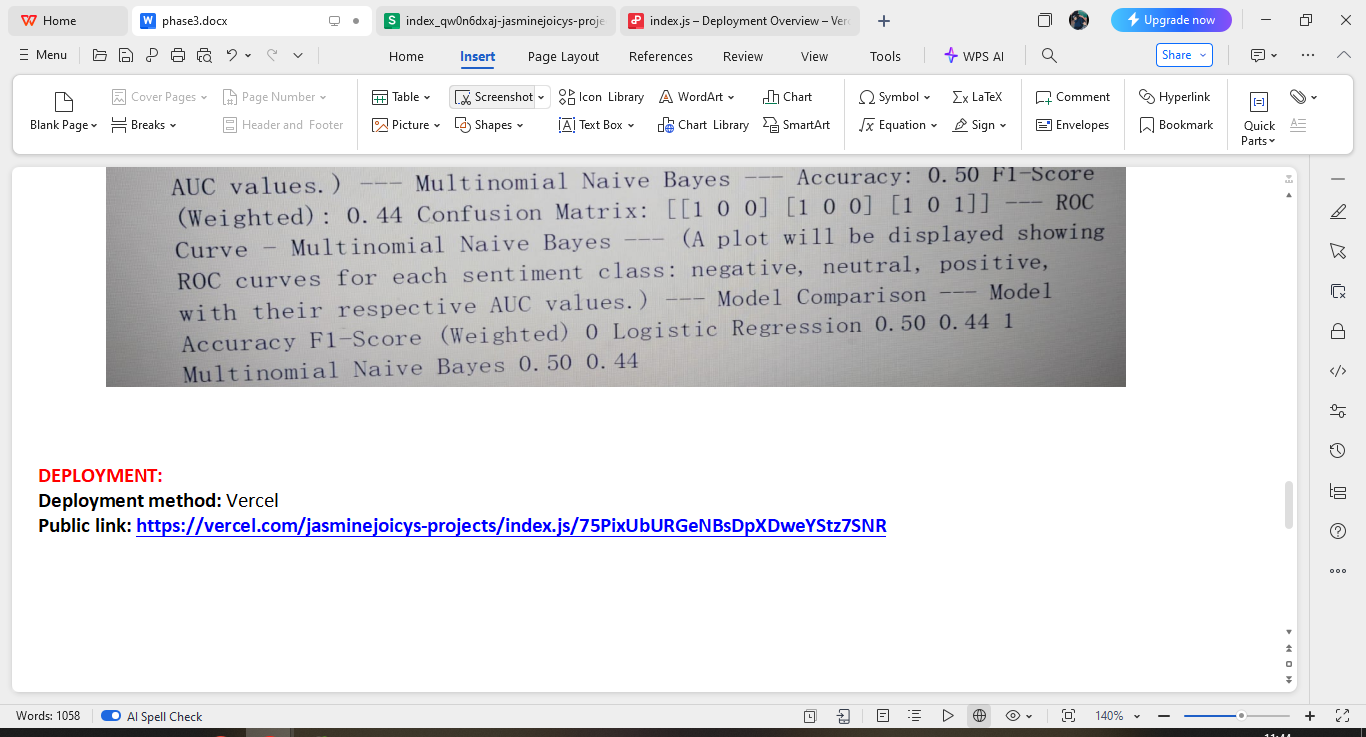
* The table compares the performance of the Logistic Regression and Multinomial Naive Bayes models based on Accuracy and F1-Score. This helps you see which model performed better on these overall metrics for the sentiment classification task.

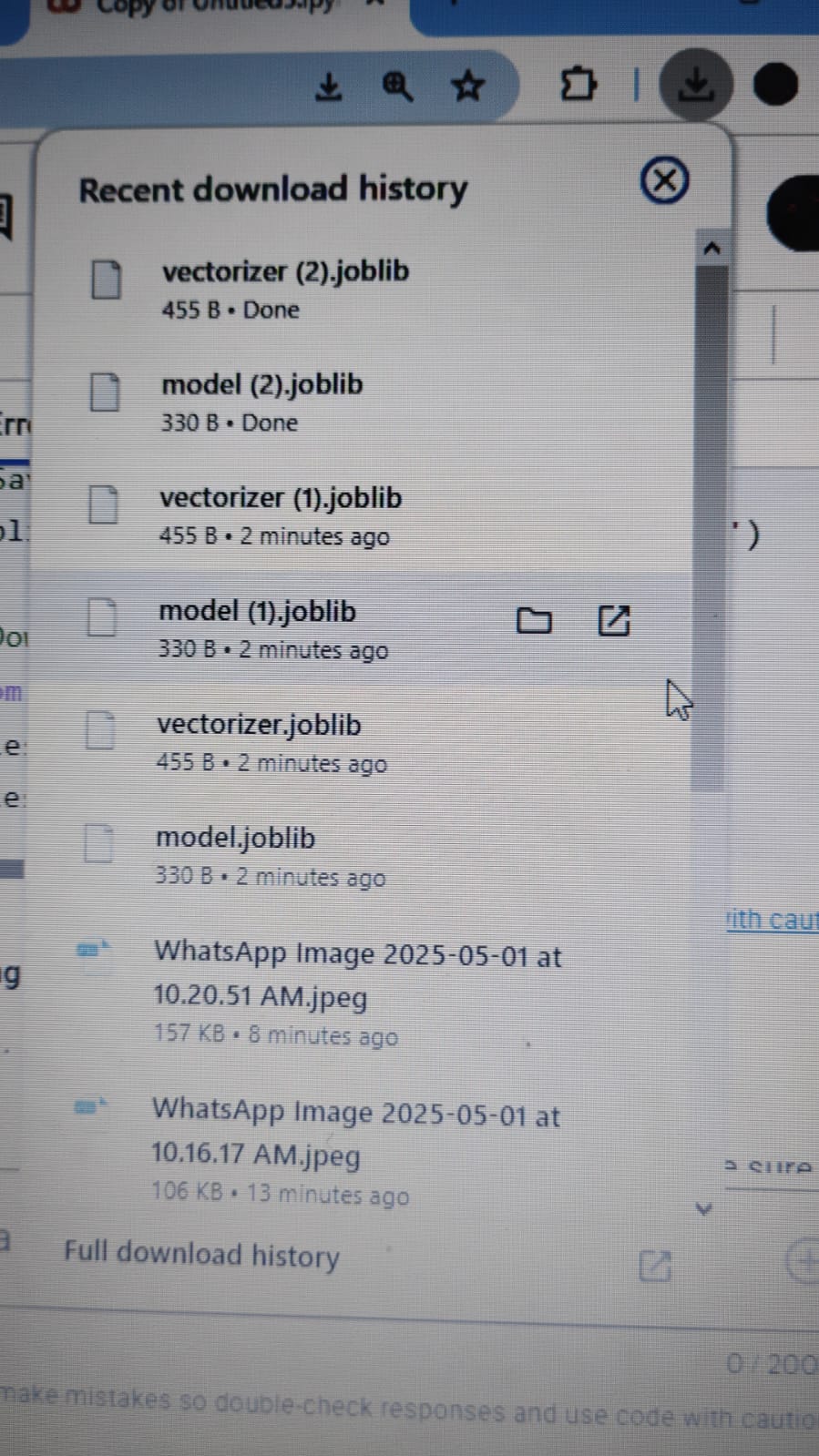


**DEPLOYMENT:**

**Deployment method:** Vercel

**Public link: <https://vercel.com/jasminejoicys-projects/index.js/75PixUbURGeNBsDpXDweYStz7SNR>**





**Source code:**

**<https://github.com/jasminejoicy/Jasmine-joicy.Data-Science.git>**

**FUTURE SCOPE:**

 **Real-time alerts:** Immediate action on critical negative feedback.

 **Multilingual analysis:** Understand regional languages accurately.

 **Proactive recommendations:** Predict and prevent future issues.

**TEAM MEMBERS AND ROLES:**

**Data cleaning:** The data cleaning process is done by P. Kavitha.

**Feature engineering:** Feature engineerting is done by D. Swathi.

**EDA:** EDA process is done by X. Mary Prakasam.

**Model Development, Documentation and reporting:** Developing the model, documentation and reporting, guiding the team for the successful execution is done by A. Jasmine Joicy.