CHAPTER 1

INTRODUCTION

1.1 AIM AND OBJECTIVES

Aim

To analyze and predict the Geographical Indication (GI) tag status of One District One Product (ODOP) products using Machine Learning techniques.

Objectives

1. To analyze product details (like sector, category, and description) and relate the products to GI tag status.
2. To predict GI status eligibility (Yes or No) for ODOP products using ML models and analyse the accuracy of the predictions.

1.2 PROBLEM STATEMENT

A study aimed to develop predictive models using Machine Learning algorithms, to classify products under ODOP scheme as eligible to obtain GI-tags for promoting local specialties from various districts. Such models can support policymakers, local entrepreneurs, and marketing agencies in identifying promising products for GI certification and strategic promotion.

Geographical Indication (GI) tags play a vital role in enhancing the market value, authenticity, and legal protection of ODOP (One District One Product) items. However, the process of obtaining a GI tag is often lengthy and resource-intensive, making it difficult to evaluate all products efficiently. With the availability of structured and textual data about ODOP products such as product name, sector, category, and description there is potential to use machine learning techniques to predict the likelihood of a product having GI status.

India is renowned for its rich cultural heritage and diverse handicrafts, agricultural produce, and traditional industries that thrive in its various districts. To promote regional specialties and local industries, the Government of India launched the One District One Product initiative. This ambitious program focuses on identifying and promoting one unique product from each district, thereby fostering economic development, encouraging entrepreneurship, and creating employment opportunities at the grassroots level. The products range from agricultural commodities, food processing items, handloom textiles, handicrafts, and marine products to manufacturing goods, reflecting the varied resource base and skills of each district.

ODOP highlights indigenous knowledge systems, crafts, and products that bear the hallmark of tradition and regional identity. A notable aspect of many ODOP products is their status pertaining to geographic indications. A Geographical Indication tag is an intellectual property right that identifies goods originating from a specific region, which possess qualities or a reputation that are due to that origin. This tag offers legal protection, market recognition, and often better pricing, thereby promoting sustainable development of regional products.

The ODOP dataset documents products district-wise, describing key attributes such as product name, category, sector, detailed description, GI status, and the responsible government ministry. The dataset reveals a vast spectrum of products, each unique in its attributes and economic potential. However, information on GI status is not uniformly available or confirmed for many entries, making it necessary to explore ways to estimate or predict this important attribute.

1.3 SOFTWARE REQUIREMENTS

The implementation of this project requires a combination of software tools and programming libraries for data processing, visualization, and model development. The details are as follows:

Operating System

- **Windows 10 / 11** - These operating systems provide a stable environment for Python-based machine learning and data analysis.

Programming Language

- **Python 3.8 or above** - Python is chosen for its simplicity, readability, and vast ecosystem of libraries that support data science, text processing, and predictive modeling.

Development Environment

- **Jupyter Notebook** – For interactive code execution and visualization.

Required Python Libraries

- **Pandas -** Data loading, cleaning, and manipulation
- **Numpy -** Numerical operations and matrix handling
- **Matplotlib**, **Seaborn-** Data visualization and graphical analysis
- **Scikit-learn -** Model building, training, and evaluation (logistic regression, accuracy, F1-score, etc.)

Documentation and Reporting Tools

- **MS Word -** For report preparation and formatting.
- **MS Excel** - For basic dataset inspection and analysis summaries.
- **PowerPoint** - For presentation of project results and visual explanations.

CHAPTER 2

LITERATURE SURVEY

[1] THE URGENCY OF GEOGRAPHICAL INDICATION AS A LEGAL PROTECTION INSTRUMENT TOWARD TRADITIONAL KNOWLEDGE IN INDONESIA

The research explores how Geographical Indications (GI) can be used as a legal tool to safeguard Indonesia's traditional knowledge. Products like coffee, pepper, and handicrafts gain their distinctive value from the region's geography, culture, and natural resources, making them important both economically and culturally. The study reviews GI laws at both national and global levels, comparing Indonesia's approach with that of countries like Ethiopia and Jamaica, and identifies issues in aligning GI protection with trademark regulations. It concludes that the current system under Indonesia's Trademark Law (Law No. 20 of 2016) provides some legal coverage but is not sufficient for fully protecting traditional knowledge. The authors suggest creating a dedicated *sui generis* legal system to offer stronger, long-term protection, boost local economic gains, and safeguard cultural heritage [1].

[2] PROTECTION OF GEOGRAPHICAL INDICATIONS AS A FORM OF CONSUMER RIGHTS PROTECTION

The study looks at Geographical Indications (GI) as a legal means of protecting both a product's identity and the rights of consumers in Indonesia. As a form of intellectual property, GIs safeguard goods whose distinct qualities come from the natural environment and human expertise of specific regions. According to Law No. 20 of 2016, GI protection is based on a formal registration process, gives collective ownership to producers, and lasts indefinitely if the product's quality and unique traits are preserved. Beyond boosting local economic development, GIs also guarantee that consumers get truthful information, consistent quality, and protection from imitation products. By connecting GI protection with consumer rights like safety, informed choices, and fair treatment, the research highlights how it benefits both producers and consumers while fostering sustainable product quality and trust in the market[2].

[3] ONE DISTRICT, ONE PRODUCT BUSINESS MODEL: A REVIEW OF DIFFERENT COUNTRIES

The paper reviews the "One District One Product" (ODOP) initiative, inspired by Japan's successful "One Village One Product" (OVOP) movement, which promotes local specialties to boost rural economies. Originally launched in Uttar Pradesh in 2018 and later expanded across

India, ODOP encourages each district to focus on a unique product—ranging from handicrafts to agricultural goods—leveraging local skills, resources, and cultural heritage. The study compares ODOP adoption in multiple countries, including Brunei, Cambodia, Indonesia, Malaysia, Myanmar, the Philippines, Vietnam, China, Thailand, Korea, and Pakistan, highlighting successes and challenges. While ODOP has contributed to job creation, poverty reduction, and the revival of traditional arts, its effectiveness varies due to differences in policy execution, stakeholder involvement, and socio-economic contexts[3].

[4] PROSPECTS AND CHALLENGES OF GEOGRAPHICAL INDICATIONS IN INDIA

The paper explores the prospects and challenges of Geographical Indications (GI) in protecting traditional knowledge and boosting economic growth in India. GI serves as an intellectual property tool that links product quality, reputation, and origin, helping preserve cultural heritage and empower rural communities. The study notes GI's potential to enhance export competitiveness, promote tourism, and increase farmers' income. However, challenges include low awareness among producers, limited institutional support, inadequate enforcement against misuse, and market accessibility barriers. To fully realize GI's benefits, the authors recommend stronger legal enforcement, producer education, marketing initiatives, and integration of GI policies into broader rural development strategies[4].

[5] GEOGRAPHICAL INDICATIONS IN INDIA: ISSUES AND CHALLENGES—AN OVERVIEW

The paper examines the role of Geographical Indications (GI) in India as a means of protecting products linked to specific regions, such as Darjeeling tea or Basmati rice. GI ensures authenticity, preserves cultural heritage, and supports rural economies by granting exclusive usage rights to authorized producers. The study outlines India's GI Act, 1999, its alignment with TRIPS Agreement provisions, and its impact on market competitiveness. While GI registration can enhance brand value and open export opportunities, challenges remain—such as inadequate enforcement against misuse, lack of producer awareness, and insufficient marketing strategies. The authors recommend capacity-building programs, stronger enforcement mechanisms, and better global promotion of registered GIs[5].

CHAPTER 3

DATA DESCRIPTION

3.1 CODEBOOK

The dataset was collected from the official government data portal (data.gov).

An overview of all the columns in the original dataset is given below

Sl.No	Column Name	Description	Data Type
1.	State	Name of the state from which the product originates.	Object
2.	Product	Product associated with the district	Object
3.	District	Name of the district associated with Product	Object
4.	LGD Code	District code	Int64
5.	Category	Classification of Product (Primary, Secondary, Tertiary)	Object
6.	Sector	Sector type (Agriculture, Handicraft, etc.)	Object
7.	Description	Details describing the products	Object
8.	GI Status	GI tag status (Yes/No)	Object
9.	Photo	Link to product photo	Object
10.	Ministry/Department	Name of the responsible government ministry/department.	Object

Table No:1 Variable Description Table for Dataset before Preprocessing

3.2 STRUCTURE OF DATA

	A	B	C	D	E	F	G	H	I	J
1	State	Product	District	LGD Code	Category	Sector	Description	GI Status	Photo	Ministry/ Department
2	Andaman and Nic	Coconut & Coconut based products	Nicobars	603	Primary	Agriculture	The Nicobar Isl	No	https://drive.google.	Ministry of Agriculture & Farmers' Welfare
3	Andaman and Nic	Fisheries/Marine products	North and Middle Andama	632	Primary	Marine	The North and I	No	https://drive.google.	Department of Fisheries
4	Andaman and Nic	Marine Products	South Andamans	602	Primary	Marine	A delight for se	No	https://drive.google.	Department of Fisheries
5	Andhra Pradesh	Coffee (Araku)	Alluri Sitarama Raju	745	Primary	Agriculture	Andhra Pradesh	Yes	https://drive.google.	Ministry of Agriculture & Farmers' Welfare
6	Andhra Pradesh	Bamboo Craft	Alluri Sitarama Raju	745	Secondary	Handicraft	Crafted with fir	No	https://drive.google.	DC Handicraft
7	Andhra Pradesh	Rock Bee Honey	Alluri Sitarama Raju	745	Tertiary	Food Processir	Rock Bee Hone	no	https://drive.google.	Ministry of Agriculture & Farmers' Welfare
8	Andhra Pradesh	Etikoppaka Lacquerware/Wooden T	Anakapalli	744	Primary	Handicraft	Anakapalli is a t	Yes	https://drive.google.	DC Handicraft
9	Andhra Pradesh	Palm Leaf	Anakapalli	744	Secondary	Agriculture	Hailing from th	No	https://drive.google.	Ministry of Agriculture & Farmers' Welfare
10	Andhra Pradesh	Payakaraopet Silk Sarees	Anakapalli	744	Tertiary	Handloom	Originating fror	No	https://drive.google.	DC Handloom
11	Andhra Pradesh	Cotton/Jean Pants	Anantapuram (Anantapur)	502	Primary	Textile	With a strong t	No	https://drive.google.	Ministry of Textile
12	Andhra Pradesh	Dharmavaram Silk Sarees	Ananthpuram (Anantapur)	502	Secondary	Handloom	Sourced from tl	Yes	https://drive.google.	DC Handloom
13	Andhra Pradesh	Cotton / Jean Pants	Annamayya (Rayachoty)	753	Primary	Textile	Cotton/Jean pa	No	https://drive.google.	Ministry of Textile
14	Andhra Pradesh	Red Sander & Wood Carvings	Annamayya (Rayachoty)	753	Secondary	Handicraft	Carved with pre	No	https://drive.google.	DC Handicraft
15	Andhra Pradesh	Raja Rani Dolls	Annamayya (Rayachoty)	753	Tertiary	Handicraft	Raja Rani Dolls,	No	https://drive.google.	DC Handicraft
16	Andhra Pradesh	Applique Work	Annamayya (Rayachoty)	753	Others	Handloom	Applique work	No	https://drive.google.	DC Handloom
17	Andhra Pradesh	Mandanapalle Silk Sarees	Annamayya (Rayachoty)	753	Others	Handloom	Mandanapalle S	No	https://drive.google.	DC Handloom
18	Andhra Pradesh	Granite Cutting and Polishing	Bapatla	750	Primary	Manufacturing	Bapatla is a sm	No	https://drive.google.	MSME
19	Andhra Pradesh	Cotton Bags	Bapatla	750	Secondary	Manufacturing	Cotton bags fro	No	https://drive.google.	MSME
20	Andhra Pradesh	Chirala Silk Sarees	Bapatla	750	Tertiary	Handloom	Chirala Silk Sare	No	https://drive.google.	DC Handloom
21	Andhra Pradesh	Venkatagiri Sarees	Chittoor	503	Primary	Handloom	Venkatagiri sar	Yes	https://drive.google.	DC Handloom
22	Andhra Pradesh	Mango Pulp	Chittoor	503	Secondary	Food processir	Chittoor is a cit	No	https://drive.google.	Ministry of Food Processing
23	Andhra Pradesh	The Glass Beads	Chittoor	503	Tertiary	Handicraft	Glass beads fro	No	https://drive.google.	DC Handicraft
24	Andhra Pradesh	Kollangantu Tie &Dye Check Lungies	Chittoor	503	Others	Textile	Kollangantu Tie	No	https://drive.google.	Ministry of Textile
25	Andhra Pradesh	Teracotta	Chittoor	503	Others	Handicraft	Teracotta from	No	https://drive.google.	DC Handicraft
26	Andhra Pradesh	Natural banana fibres	Dr BR Ambedkar Konaseen	747	Others	Manufacturing	Natural banana	No	https://drive.google.	MSME
27	Andhra Pradesh	Bandaru Lanka Sarees	Dr BR Ambedkar Konaseen	747	Others	Handloom	Bandaru Lanka	No	https://drive.google.	DC Handloom

3.3 DATA PREPROCESSING

Checked the overall shape, data types, and sample records using df.shape, df.dtypes, and df.head().

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df = pd.read_excel("One District One Product.xlsx")
print("Shape of dataset:", df.shape)
```

```
Shape of dataset: (1241, 10)
```

```
print("\nData Types:\n", df.dtypes)
```

```
Data Types:
 State                 object
Product                object
District               object
LGD Code                int64
Category               object
Sector                 object
Description            object
GI Status              object
Photo                  object
Ministry/ Department   object
dtype: object
```

```
print("\nSample Data:\n", df.head())
```

```
Sample Data:
                         State                            Product  \
0  Andaman and Nicobar Islands  Coconut & Coconut based products
1  Andaman and Nicobar Islands         Fisheries/Marine products
2  Andaman and Nicobar Islands                    Marine Products
3               Andhra Pradesh                     Coffee (Araku)
4               Andhra Pradesh                       Bamboo Craft

                  District  LGD Code   Category       Sector  \
0                 Nicobars       603    Primary  Agriculture
1  North and Middle Andaman       632    Primary       Marine
2            South Andamans       602    Primary       Marine
3      Alluri Sitarama Raju       745    Primary  Agriculture
4      Alluri Sitarama Raju       745  Secondary   Handicraft

                                        Description GI Status  \
0  The Nicobar Islands are known for their abunda...        No
1  The North and Middle Andaman islands, are know...        No
2  A delight for seafood enthusiasts, South Andam...        No
3  Andhra Pradesh is one of the major coffee prod...       Yes
4  Crafted with finesse in Andhra Pradesh's Allur...        No

                                               Photo  \
0  https://drive.google.com/file/d/1ZKef7mLS4iub1...
1  https://drive.google.com/file/d/1caPuTWCA35HHL...
2  https://drive.google.com/file/d/15DIayCr05l903...
3  https://drive.google.com/file/d/18ptepDrMFsm5l...
4  https://drive.google.com/file/d/1HT-KGuEaGPfvf...

                        Ministry/ Department
0  Ministry of Agriculture & Farmers' Welfare
1                     Department of Fisheries
2                     Department of Fisheries
3  Ministry of Agriculture & Farmers' Welfare
4                               DC Handicraft
```

Verified missing values and summary statistics using df.isnull().sum() and df.describe() to understand data quality and structure.

```
print("\nDescription of Dataset:\n", df.describe())
```

```
Description of Dataset:
           LGD Code
count  1241.000000
mean    394.124899
std     218.368263
min       1.000000
25%     207.000000
50%     401.000000
75%     569.000000
max     782.000000
```

```
print("\nMissing Values:\n", df.isnull().sum())
```

```
Missing Values:
 State                  0
Product                 0
District                0
LGD Code                0
Category                0
Sector                  0
Description             0
GI Status               0
Photo                   0
Ministry/ Department    0
dtype: int64
```

```
print("\nInformation of Dataset:\n",df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1241 entries, 0 to 1240
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   State                 1241 non-null   object
 1   Product               1241 non-null   object
 2   District              1241 non-null   object
 3   LGD Code              1241 non-null   int64
 4   Category              1241 non-null   object
 5   Sector                1241 non-null   object
 6   Description           1241 non-null   object
 7   GI Status             1241 non-null   object
 8   Photo                 1241 non-null   object
 9   Ministry/ Department  1241 non-null   object
dtypes: int64(1), object(9)
memory usage: 97.1+ KB

Information of Dataset:
 None
```

Dropped non-essential columns “LGD Code” and “Photo” as they do not contribute to GI Tag prediction.

```
df_clean = df.drop(columns=["LGD Code", "Photo"])
df_clean
```

	State	Product	District	Category	Sector	Description	GI Status	Ministry/ Department
0	Andaman and Nicobar Islands	Coconut & Coconut based products	Nicobars	Primary	Agriculture	The Nicobar Islands are known for their abunda...	No	Ministry of Agriculture & Farmers' Welfare
1	Andaman and Nicobar Islands	Fisheries/Marine products	North and Middle Andaman	Primary	Marine	The North and Middle Andaman islands, are know...	No	Department of Fisheries
2	Andaman and Nicobar Islands	Marine Products	South Andamans	Primary	Marine	A delight for seafood enthusiasts, South Andam...	No	Department of Fisheries
3	Andhra Pradesh	Coffee (Araku)	Alluri Sitarama Raju	Primary	Agriculture	Andhra Pradesh is one of the major coffee prod...	Yes	Ministry of Agriculture & Farmers' Welfare
4	Andhra Pradesh	Bamboo Craft	Alluri Sitarama Raju	Secondary	Handicraft	Crafted with finesse in Andhra Pradesh's Allur...	No	DC Handicraft

Converted categorical text values into a consistent format by applying str.strip() and str.lower() to remove extra spaces and unify letter cases.

Mapped ‘yes’ and ‘no’ values in the GI Status column to standardized labels ‘Yes’ and ‘No’ for binary classification.

```
df_clean['GI Status'] = df_clean['GI Status'].str.strip().str.lower().map({
    'yes': 'Yes',
    'no': 'No'
})

# Confirm the changes
print(df_clean['GI Status'].value_counts())
```

```
GI Status
No     985
Yes    256
Name: count, dtype: int64
```

Cleaned the Category column by mapping text values to proper categories: Primary, Secondary, and Tertiary.

```
df_clean['Category'] = df_clean['Category'].str.strip().str.lower().map({
    'primary': 'Primary',
    'secondary': 'Secondary',
    'tertiary': 'Tertiary'
})

# Confirm the changes
print(df_clean['Category'].value_counts())
```

```
Category
Primary      776
Secondary    314
Tertiary     103
Name: count, dtype: int64
```

Created a new column ‘Sector_clean’ to handle inconsistent spellings and capitalization in the Sector column.

Mapped similar sector names like 'handlooms' → 'Handloom', 'food processing' → 'Food Processing', etc.

Filled missing mappings with original values using fillna().

```
df_clean['Sector'].value_counts()
```

```
Sector
Agriculture        418
Handicraft         251
Manufacturing      173
Food Processing    115
Textile            103
Handloom           100
Marine              32
Others              29
Dairy                9
Food processing      7
dairy                1
textile              1
handicraft           1
handlooms            1
Name: count, dtype: int64
```

```python
df_clean['Sector_clean'] = df_clean['Sector'].str.strip().str.lower()
sector_mapping = {
    'handicraft': 'Handicraft',
    'dairy': 'Dairy',
    'food processing': 'Food Processing',
    'textile': 'Textile',
    'handlooms': 'Handloom',
    'handloom': 'Handloom'
}

df_clean['Sector_clean'] = df_clean['Sector_clean'].map(sector_mapping).fillna(df_clean['Sector'])
print(df_clean['Sector_clean'].value_counts())
```

```
Sector_clean
Agriculture        418
Handicraft         252
Manufacturing      173
Food Processing    122
Textile            104
Handloom           101
Marine              32
Others              29
Dairy               10
Name: count, dtype: int64
```

```python
df_clean['Sector_clean'].unique()
```

```
array(['Agriculture', 'Marine', 'Handicraft', 'Food Processing',
       'Handloom', 'Textile', 'Manufacturing', 'Others', 'Dairy'],
      dtype=object)
```

CHAPTER 4

EXPLORATORY DATA ANALYSIS

4.1 NEED AND IMPORTANCE

Exploratory Data Analysis is one the basic and essential steps in any data analysis projects.EDA refers to the method of studying and exploring the data in order to discover patterns, locate outliers and find the relationship between the variables. EDA involves examining the information for errors, lacking values, and inconsistencies.

EDA is an approach to understand the data by making the summarization and visual representation on the data. EDA will give better features to be used to find more useful insight from the data. The main aim of the EDA process is to use statistical techniques to efficiently summarize and visualize a better view of data, and find values about the importance of the data, its quality, and derive the new perspective and the suggestion of our analysis.

EDA process is iterative in nature because it helps to make assumptions on the first look of the data and then extract some useful insights from that data to build the machine learning models. It helps to make use of visualization techniques to preview the model results and tune them according to the applications.

4.2 UNIVARIATE ANALYSIS

Univariate Analysis is a type of data visualization where we visualize only a single variable at a time. Univariate Analysis helps us to analyze the distribution of the variable present in the data so that we can perform further analysis.

4.2.1 UNIVARIATE ANALYSIS FOR CATEGORICAL DATA

The chart shows the number of products in each category, with Primary having the highest count, followed by Secondary and Tertiary.

```
# Category
plt.figure(figsize=(8,4))
sns.countplot(y=df_clean['Category'], order=df_clean['Category'].value_counts().index,
              hue=df_clean['Category'],palette="coolwarm")
plt.title("Products by Category")
plt.xlabel("Count")
plt.ylabel("Category")
plt.show()
```

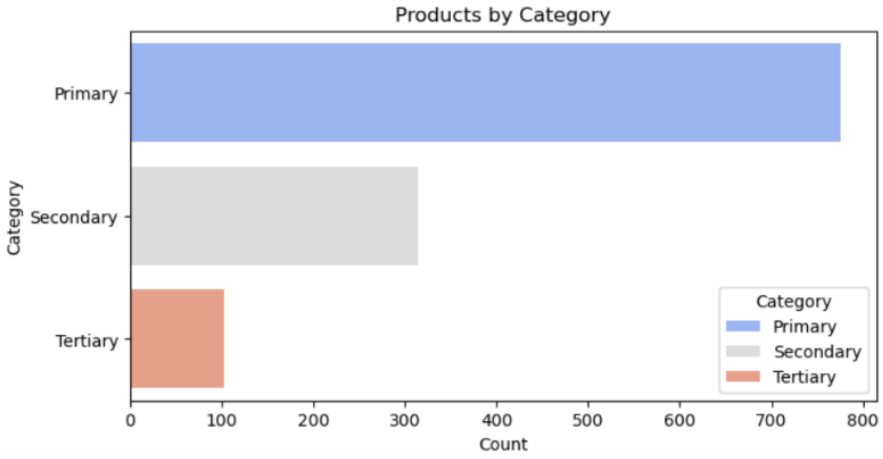


Fig. No. 1 Count of Products by Category using a Horizontal Bar Plot

The chart shows the number of products in each sector, with Agriculture having the most and Dairy the least.

```
# Sector
sector_counts = df_clean['Sector_clean'].value_counts()
sns.barplot(x=sector_counts.values, y=sector_counts.index,hue=sector_counts.index,palette="viridis")
plt.title("Products by Sector")
plt.xlabel("Total Count")
plt.ylabel("Sector")
plt.show()

print(df_clean['Sector_clean'].value_counts())
```

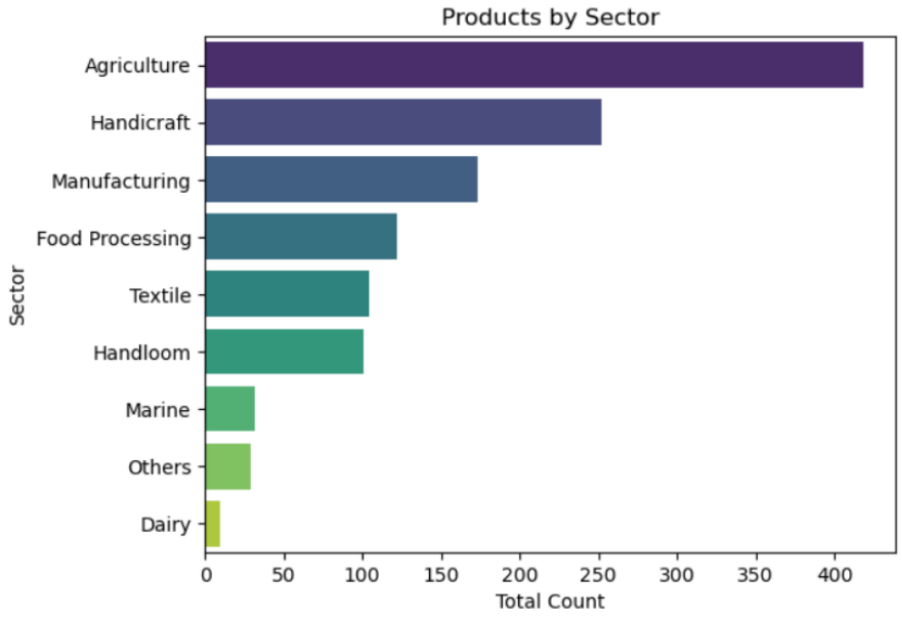


Fig. No. 2 Distribution of ODOP Products by Sector using a Horizontal Bar Plot

This bar graph shows the number of ODOP (One District One Product) items from each state in India. Uttar Pradesh has the highest number of products, followed by Andhra Pradesh, Gujarat, Bihar, and Maharashtra. Smaller states and union territories like Lakshadweep and Daman & Diu have fewer products.

```
# State
plt.figure(figsize=(12,6))
sns.countplot(y=df_clean['State'], order=df_clean['State'].value_counts().index, hue=df_clean['State'], palette="magma")
plt.title("States with Most Products")
plt.yticks(rotation=0,fontsize=6,fontweight='bold')
plt.xlabel("Count")
plt.ylabel("State")
plt.figure(figsize=(12, 14))
plt.tight_layout()
plt.show()

print(df_clean['State'].value_counts())
```

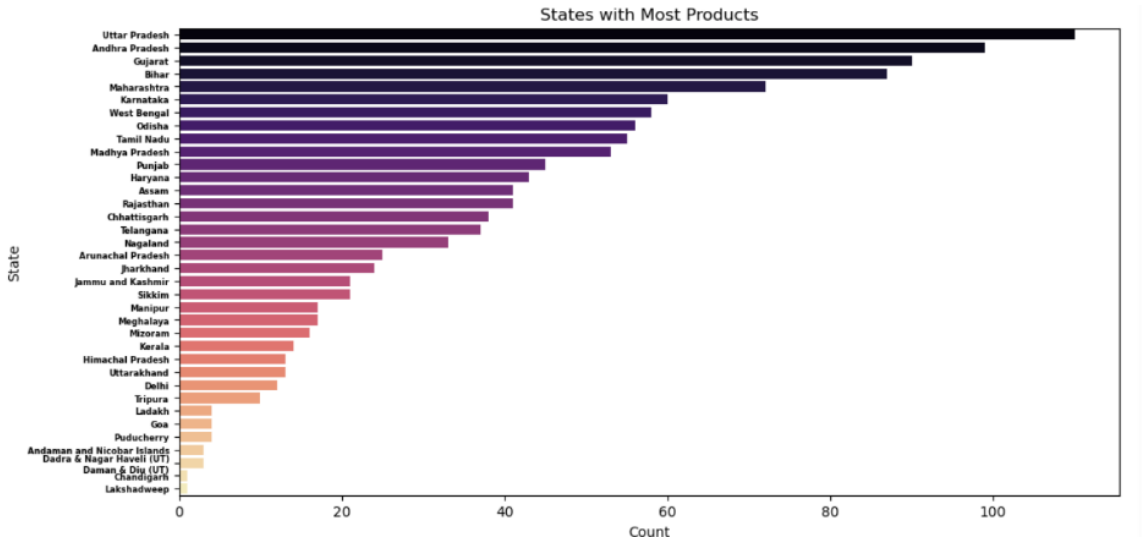


Fig. No. 3 State-wise Distribution of ODOP Products in India

4.2.2 UNIVARIATE ANALYSIS FOR NUMERICAL DATA

This chart shows that 79.4% of products have no GI status, while only 20.6% are GI certified.

```
# Count of each GI Status
gi_counts = df_clean['GI Status'].value_counts()

plt.figure(figsize=(4,4))
plt.pie(gi_counts, labels=['No GI', 'Yes GI'], autopct='%1.1f%%', startangle=90)
plt.legend(loc="upper left")
plt.title("Distribution of GI Status")
plt.axis('equal')
plt.show()
```

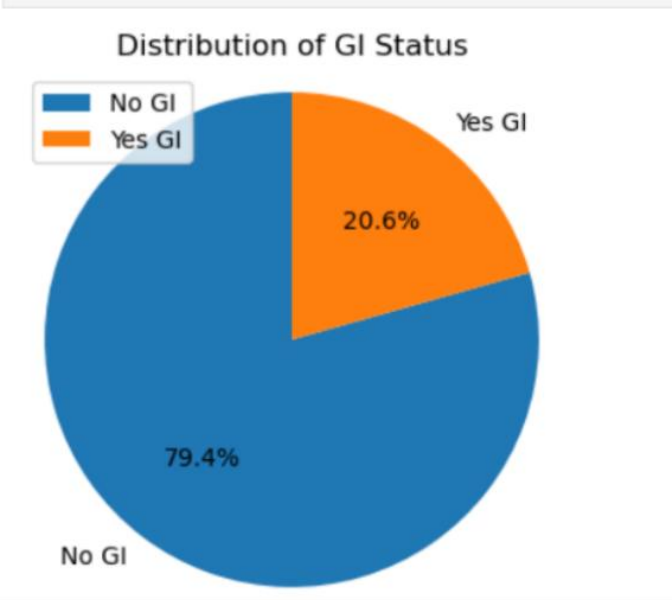


Fig. No. 4 Distribution of GI Status using a Pie Chart

4.3 BIVARIATE ANALYSIS

Bivariate analysis is the simultaneous analysis of two variables. It explores the concept of the relationship between two variables whether there exists an association and the strength of this association or whether there are differences between two variables and the significance of these differences.

4.3.1 BIVARIATE ANALYSIS FOR CATEGORICAL DATA

This chart compares GI Status across different sectors, showing that most sectors have more products without GI certification than with it.

```
sns.countplot(data=df_clean, x='Sector_clean', hue='GI Status', palette='magma')
plt.title("GI Status vs Sector", fontsize=14)
plt.xlabel("Sector")
plt.ylabel("Count")
plt.xticks(rotation=45, ha='right')
plt.legend(title='GI Status', labels=['No', 'Yes'])
plt.tight_layout()
plt.show()
```

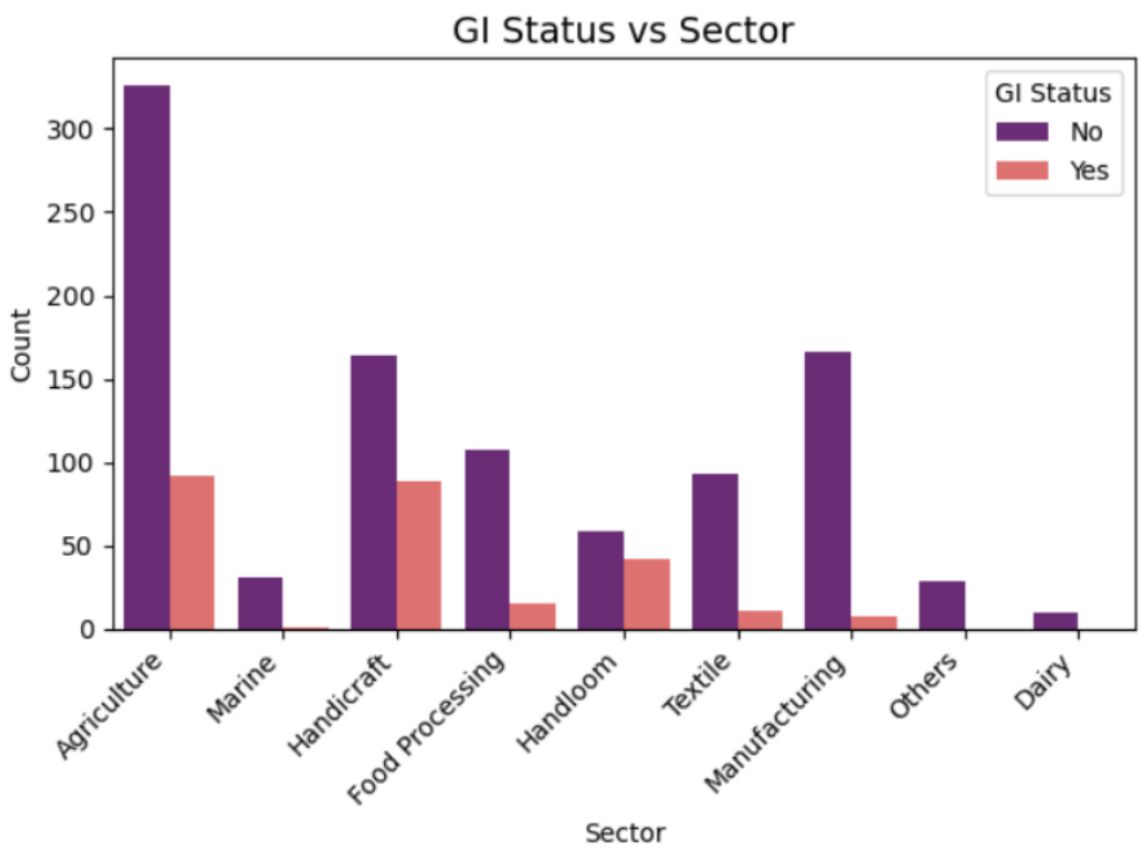


Fig. No. 5 Count of GI Status by Sector using a Clustered Bar Plot

This chart shows that most products in the top 10 states are not GI certified, though Tamil Nadu and Uttar Pradesh have relatively more GI-certified items.

```
top_states = df_clean['State'].value_counts().head(10).index
sns.countplot(data=df_clean[df_clean['State'].isin(top_states)], x='State', hue='GI Status', palette="viridis")
plt.title("GI Status vs Top 10 States")
plt.xticks(rotation=45, ha="right")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```

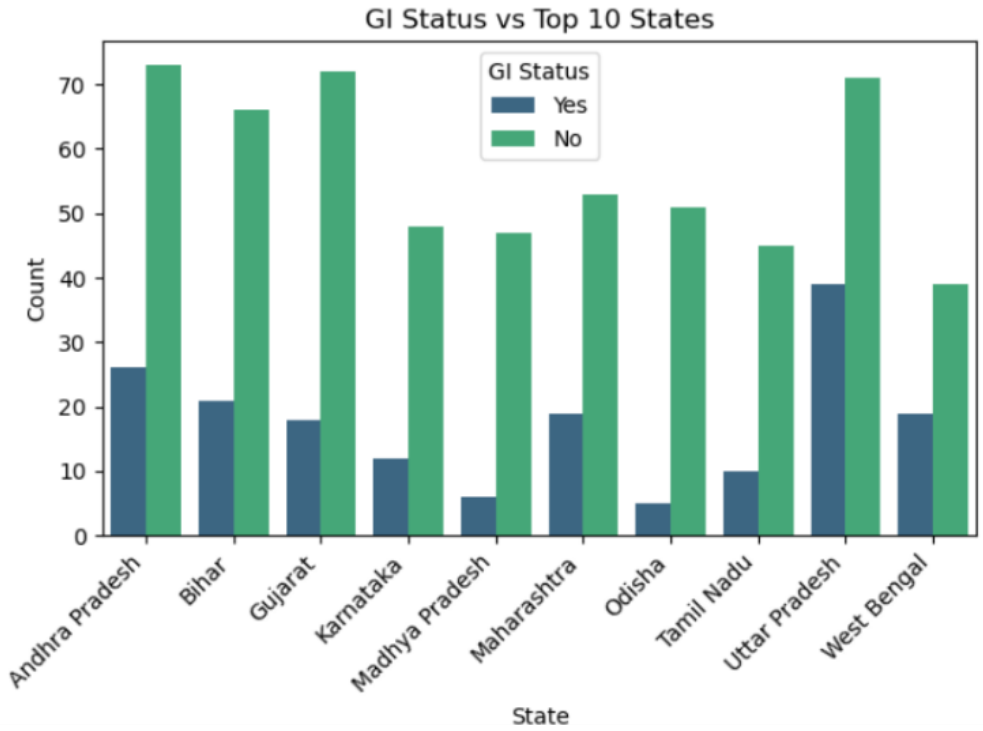


Fig. No. 6 Count of GI Status by State using a Clustered Bar Plot

The chart illustrates that individuals with 'No' GI status consistently outnumber those with 'Yes' GI status across all categories, with the Primary category showing the highest overall count.

```
#GI Status vs Category
plt.figure(figsize=(10,5))
sns.countplot(data=df_clean, x='Category', hue='GI Status', palette="coolwarm")
plt.title("GI Status vs Category")
plt.xticks(rotation=45, ha="right")
plt.ylabel("Count")
plt.show()

print(pd.crosstab(df_clean['Category'],df_clean['GI Status']))
```

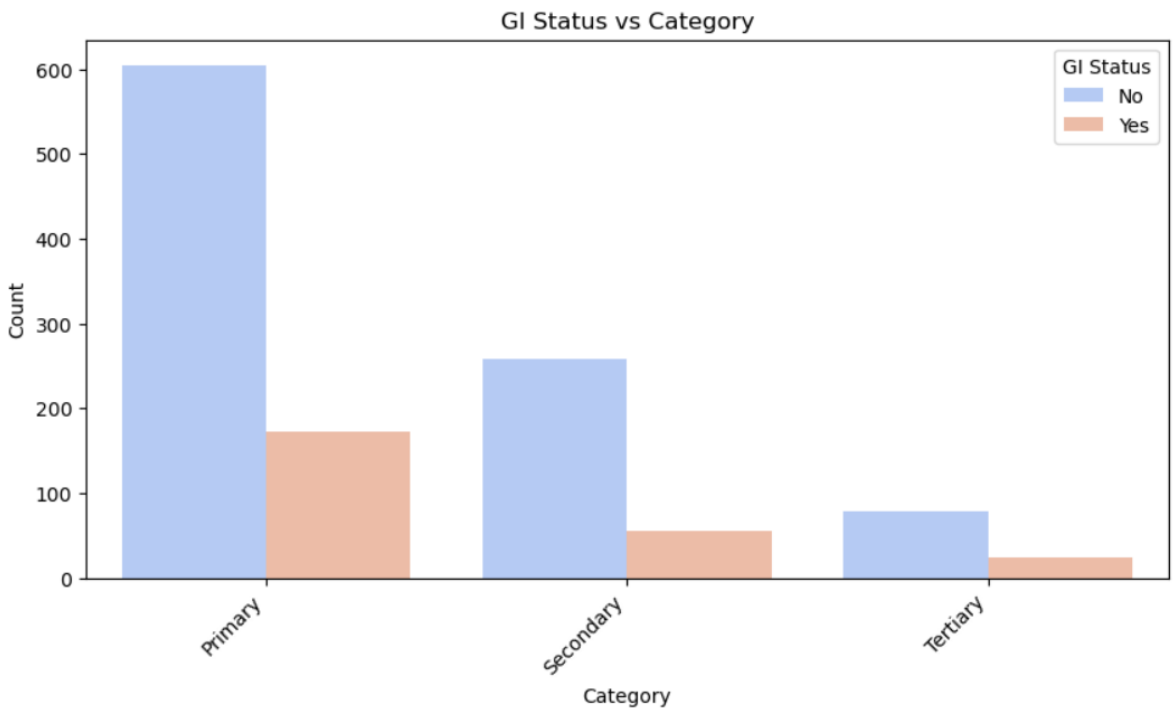


Fig. No. 7 Count of GI Status by Category using a Clustered Bar Plot

The chart highlights significant variation in category distribution across Indian states, with bubble size and color indicating that Primary education levels dominate in most regions, while Tertiary representation remains comparatively lower.

```
#State vs Category
grouped = df_clean.groupby(['State', 'Category']).size().reset_index(name='Count')

plt.figure(figsize=(12,6))
sns.scatterplot(data=grouped, x='State', y='Category', size='Count', sizes=(50, 500), hue='Count', palette='coolwarm')
plt.title("State vs Category")
plt.xticks(rotation=90)
plt.tight_layout()
plt.legend(loc='upper left',bbox_to_anchor=(1.05, 1),borderaxespad=0)
plt.show()
```

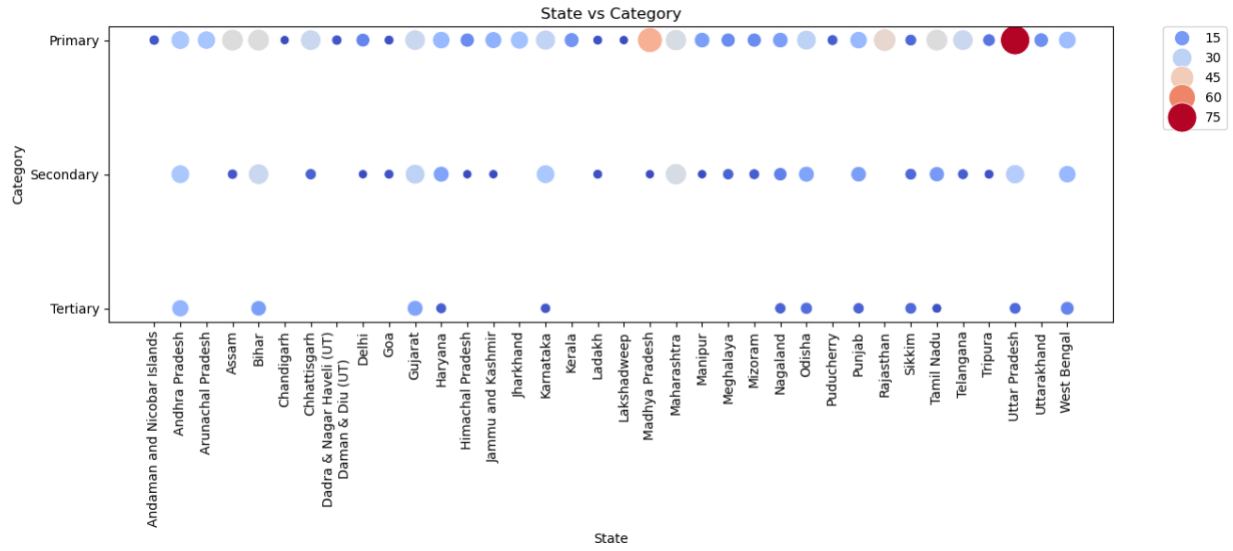


Fig. No. 8 Distribution of Category by State using a Bubble Chart

4.4 MULTIVARIATE ANALYSIS

It is an extension of bivariate analysis which means it involves multiple variables at the same time to find correlation between them. Multivariate Analysis is a set of statistical models that examine patterns in multidimensional data by considering, at once, several data variables.

4.4.1 MULTIVARIATE ANALYSIS FOR CATEGORICAL DATA

The heatmap shows how GI Status varies across sector-category combinations, with Agriculture-Primary having the most non-GI products and Handicraft-Secondary showing the highest count of GI-certified items.

```
plt.figure(figsize=(10,6))
cross_tab = pd.crosstab([df_clean['Sector_clean'], df_clean['Category']], df_clean['GI Status'])
sns.heatmap(cross_tab, annot=True, fmt='d', cmap='YlGnBu')
plt.title("Heatmap: Sector & Category vs GI Status")
plt.show()
```

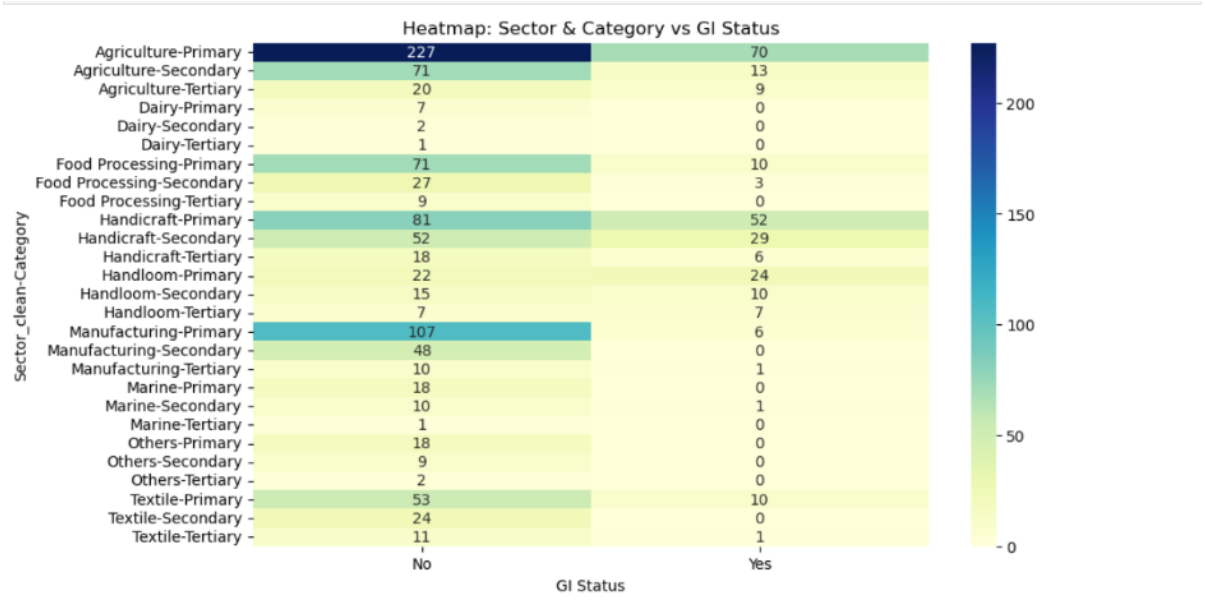


Fig. No. 9 Heatmap of GI Status Counts by Sector and Product Category

CHAPTER 5

METHODOLOGY

WORKFLOW DIAGRAM OF PREDICTION

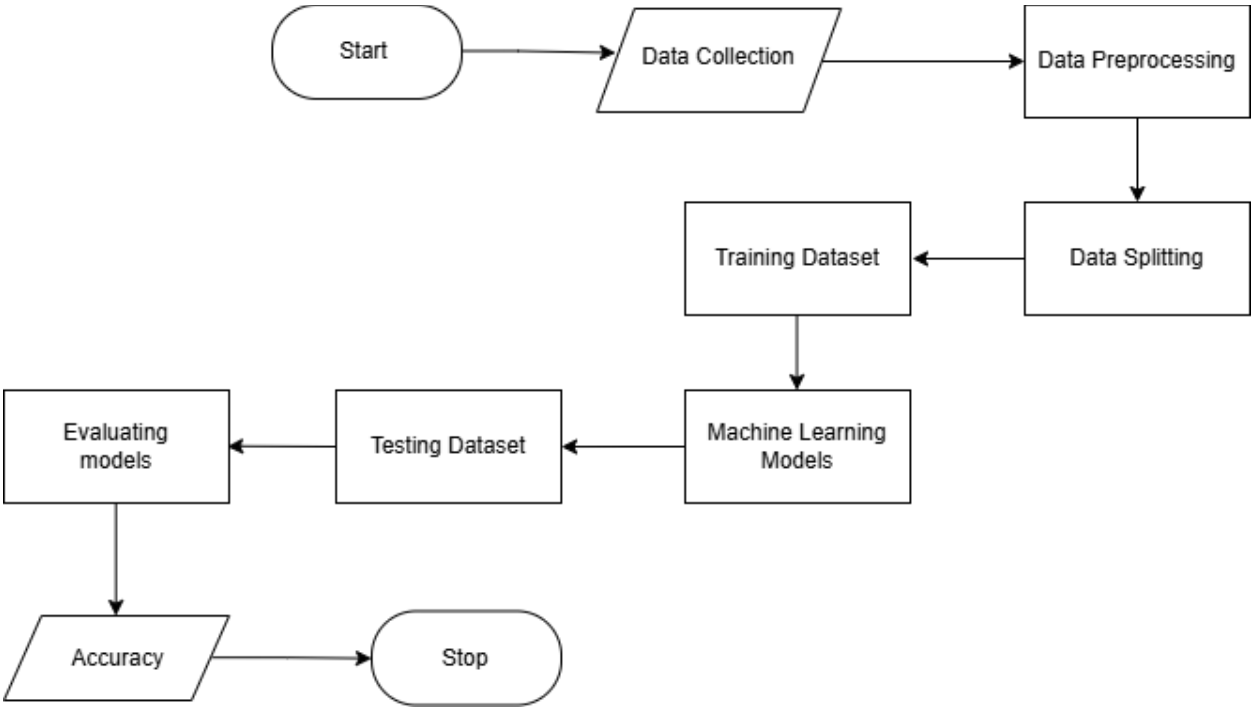


Fig. No. 10 Workflow diagram of prediction

5.1 MODEL 1- Logistic Regression

Logistic Regression is a supervised machine learning algorithm used for binary classification problems. In this project, it predicts whether an ODOP product has a Geographical Indication (GI) tag (*Yes* or *No*). The algorithm models the probability of a product belonging to the "GI" class based on input features such as sector, category, and textual description. It applies the sigmoid function to map predictions between 0 and 1. Logistic Regression was chosen as the baseline model due to its interpretability, simplicity, and efficiency with linearly separable data. Feature importance from this model helps in understanding which variables most influence GI status prediction.

5.2 MODEL 2 - Random Forest Classifier

Random Forest is an ensemble learning technique that constructs multiple decision trees and combines their outputs to improve accuracy and reduce overfitting. Each tree in the forest is trained on a random subset of data and features, and the final prediction is made through majority voting. In this project, Random Forest was used to capture non-linear relationships among features and handle both categorical and textual feature sets effectively. The model's feature importance metrics help identify the key predictors influencing GI classification. Random Forest provides robustness and high predictive performance, especially when dealing with complex data structures.

5.3 MODEL 3 - Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is an advanced boosting algorithm known for its speed, regularization, and high accuracy. It builds trees sequentially, where each new tree corrects the errors made by previous ones. XGBoost applies gradient descent optimization to minimize classification error and includes regularization terms to prevent overfitting. For this project, XGBoost was used to improve performance over simpler ensemble methods by optimizing both bias and variance. It efficiently handles heterogeneous data (numerical, categorical, and text-derived features) and provides feature importance rankings for interpretability.

5.4 MODEL 4 - K-Nearest Neighbors

The K-Nearest Neighbors (KNN) algorithm is a non-parametric method that classifies a new data point based on the majority class of its *K* nearest neighbors in the feature space. It works on the principle that similar products (with similar features and descriptions) are likely to have the same GI status. In this project, the Euclidean distance metric was used to determine proximity between products. KNN is simple and intuitive but sensitive to feature scaling and the choice of *K*. It serves as a useful benchmark model for comparing performance against more complex algorithms like Random Forest and XGBoost.

5.5 MODEL 5 - Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. VM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

CHAPTER 6

IMPLEMENTATION OF THE ALGORITHMS

6.1 IMPLEMENTATION AND RESULTS

Text data was cleaned by converting to lowercase, removing special characters, and combining product, sector, and category fields to create a unified feature for analysis

```
# Text preprocessing function
def preprocess_text(text):
    text = str(text).lower().strip()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    return text

# Clean and combine text features
df_clean['Product_clean'] = df_clean['Product'].apply(preprocess_text)
df_clean['Combined_Text'] = df_clean['Product_clean'] + " " + df_clean['Sector_clean'] + " " + df_clean['Category']
```

Rows with missing GI Status were removed, and the target variable was encoded as binary—assigning 1 to 'Yes' and 0 to 'No'—to prepare the data for classification

```
# Drop rows with missing GI Status
df_clean = df_clean[df_clean['GI Status'].notna()]

# Features and target
X_text = df_clean['Combined_Text'].astype(str)
y = df_clean['GI Status'].map({'Yes': 1, 'No': 0})
```

Text features were transformed using TF-IDF vectorization, and the dataset was split into training and testing sets to evaluate model performance.

```
# TF-IDF vectorization
vectorizer = TfidfVectorizer()
X_vectorized = vectorizer.fit_transform(X_text)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_vectorized, y, test_size=0.2, random_state=42)
```

1. Logistic Regression

The first algorithm used in this project is Logistic Regression. A logistic regression model was trained and evaluated, achieving 76% accuracy, with strong precision for non-GI products and moderate recall for GI-certified items, as shown in the classification report and confusion matrix.

```
# Logistic Regression model
lr_model = LogisticRegression(max_iter=1000, class_weight='balanced')
lr_model.fit(X_train, y_train)

# Predictions
y_pred = lr_model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7590361445783133

Confusion Matrix:
[[154  39]
 [ 21  35]]

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.80      0.84       193
           1       0.47      0.62      0.54        56

    accuracy                           0.76       249
   macro avg       0.68      0.71      0.69       249
weighted avg       0.79      0.76      0.77       249
```

2. Random Forest Classifier

The second algorithm used in this project is Random Forest Classifier. The Random Forest Classifier achieved an accuracy of 81%, outperforming logistic regression in overall prediction. While it showed excellent precision and recall for non-GI products.

```
# Random Forest Classifier
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)

# Predictions
y_pred = rf_model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8112449799196787

Confusion Matrix:
[[184   9]
 [ 38  18]]

Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.95      0.89       193
           1       0.67      0.32      0.43        56

    accuracy                           0.81       249
   macro avg       0.75      0.64      0.66       249
weighted avg       0.79      0.81      0.78       249
```

3. Gradient Boosting (XGBoost)

The third algorithm used in this project is XGBoost classifier. The XGBoost classifier achieved an accuracy of 85%, demonstrating strong predictive performance. It showed high precision for both classes and moderate recall for GI-certified products.

```
# XGBoost Classifier
xgb_model = XGBClassifier(n_estimators=200,learning_rate=0.1,max_depth=6,
                          random_state=42,eval_metric='logloss')
xgb_model.fit(X_train, y_train)

# Predictions
y_pred = xgb_model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8473895582329317

Confusion Matrix:
[[183  10]
 [ 28  28]]

Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.95      0.91       193
           1       0.74      0.50      0.60        56

    accuracy                           0.85       249
   macro avg       0.80      0.72      0.75       249
weighted avg       0.84      0.85      0.84       249
```

4. K-Nearest Neighbors

The fourth algorithm used in this project is K-Nearest Neighbor Classifier. The KNN classifier achieved an accuracy of approximately 80%, showing strong recall for non-GI products and high precision for GI-certified items. The model effectively captured neighborhood patterns in the data.

```
# K-Nearest Neighbors
knn_model = KNeighborsClassifier(n_neighbors=5)
knn_model.fit(X_train, y_train)

# Predictions
y_pred = knn_model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7991967871485943

Confusion Matrix:
[[188   5]
 [ 45  11]]

Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.97      0.88       193
           1       0.69      0.20      0.31        56

    accuracy                           0.80       249
   macro avg       0.75      0.59      0.59       249
weighted avg       0.78      0.80      0.75       249
```

5. Support Vector Machine

The fifth algorithm used in this project is Support Vector Machine. The SVM classifier achieved an accuracy of approximately 79%, with strong performance in identifying non-GI products.

```
#Support Vector Machine (SVM)
svm_model = SVC(kernel='rbf', C=1.0, gamma='scale', class_weight='balanced', probability=True)
svm_model.fit(X_train, y_train)

# Predictions
y_pred = svm_model.predict(X_test)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.7871485943775101

Confusion Matrix:
[[171  22]
 [ 31  25]]

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.89      0.87       193
           1       0.53      0.45      0.49        56

    accuracy                           0.79       249
   macro avg       0.69      0.67      0.68       249
weighted avg       0.78      0.79      0.78       249
```

PREDICTION PROGRAM

Text was cleaned and combined to prepare it for model input.

```
import re
def preprocess_text(text):
    text = str(text).lower().strip()
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    return text

# Ensuring no NaN in Product, Sector, Category
df_clean['Product'] = df_clean['Product'].fillna("Unknown")
df_clean['Sector_clean'] = df_clean['Sector_clean'].fillna("Unknown")
df_clean['Category'] = df_clean['Category'].fillna("Unknown")

# Preprocess and combine
df_clean['Product_clean'] = df_clean['Product'].apply(preprocess_text)
df_clean['Combined_Text'] = df_clean['Product_clean'] + " " + df_clean['Sector_clean'] + " " + df_clean['Category']
```

Text data was converted into numerical features using TF-IDF vectorization for model training.

```
# vectorizer used during training
vectorizer = TfidfVectorizer()
X_vectorized = vectorizer.fit_transform(X_text)
```

Predictions from multiple models were aggregated using a majority vote with a lowered threshold, resulting in a final classification of products into 'Yes GI' and 'No GI' categories.

```
# Predictions from all models
predictions = np.zeros((X_vectorized.shape[0], len(models)))

for i, (name, model) in enumerate(models.items()):
    predictions[:, i] = model.predict(X_vectorized)

# Apply a  lower threshold for "Yes GI"
yes_threshold = 0.1  # Default was 0.5

majority_vote = np.apply_along_axis(
    lambda x: 1 if (sum(x) / len(x)) >= yes_threshold else 0,
    axis=1,
    arr=predictions
)

# Adding predicted GI status to DataFrame
df_clean['Predicted_GI_Status'] = np.where(majority_vote == 1, 'Yes GI', 'No GI')

# Overall distribution of predictions
print(df_clean['GI Status'].value_counts())
print(df_clean['Predicted_GI_Status'].value_counts())
```

```
GI Status
No     985
Yes    256
Name: count, dtype: int64
Predicted_GI_Status
No GI     845
Yes GI    396
Name: count, dtype: int64
```

A sample of predicted results was displayed, showing product details along with actual and predicted GI status for verification.

```
print(df_clean[['Product_clean' ,'Category','Sector_clean','GI Status', 'Predicted_GI_Status']].sample(10))
```

```
                          Product_clean   Category     Sector_clean  \
62    bandar type resham art silk sarees  Secondary         Handloom
303             readymade clothesgarments    Primary          Textile
620                      ratlami namkeen    Primary  Food Processing
18                   chirala silk sarees   Tertiary         Handloom
1211        kids wear readymade garments  Secondary          Textile
48              kalamkari block printing   Tertiary         Handloom
786                             anthurium  Secondary      Agriculture
149                                ginger    Primary      Agriculture
459                           trout fish    Primary           Marine
78                 ponduru cotton sarees    Primary          Textile

     GI Status Predicted_GI_Status
62          No              Yes GI
303         No               No GI
620        Yes               No GI
18          No              Yes GI
1211        No               No GI
48          No              Yes GI
786         No               No GI
149        Yes              Yes GI
459         No               No GI
78          No               No GI
```

6.2 COMPARISON OF MODEL

Sl No	Model Implemented	Accuracy
1	XGBoost Classifier	84.7%
2	Random Forest Classifier	81.1%
3	K-Nearest Neighbor Classifier	79.9%
4	Support Vector Machine	78.7%
5	Logistic Regression	76.0%

Table 2 : Accuracy Table for Algorithm

From the above table, it can be inferred that XGBoost Classifier algorithm returns the best accuracy value i.e., 84.7%. Hence it can be concluded that XGBoost Classifier algorithm is the best algorithm to predict the Geographical Indication (GI) tag for One District One Product (ODOP) Products.

CHAPTER 7

GRAPHICAL USER INTERFACE

The **Graphical User Interface (GUI)** serves as the front-end component of the project, allowing users to interact easily with the machine learning model and visualize its results without needing to work directly with the code. The interface provides a simple, intuitive, and user-friendly experience for uploading data, running predictions, and viewing outputs.

7.1 IMPLEMENTATION OF THE GUI

The GUI was developed using Python's Tkinter which enables quick creation of interactive and responsive interfaces. The design ensures that even non-technical users, such as government officials, researchers, or local entrepreneurs, can make effective use of the prediction model.

```
#tkinter
import tkinter as tk
from tkinter import messagebox
import pandas as pd
```

The function get_all_rows_info() retrieves all matching entries for a given product from the cleaned dataset, returning structured details including sector, category, and predicted GI status

```
def get_all_rows_info(product_name):
    # Preprocess product like training
    product_clean = str(product_name).lower().strip()

    # Find all matching rows
    rows = df_clean[df_clean['Product_clean'] == product_clean]

    if not rows.empty:
        info_list = []
        for idx, row in rows.iterrows():
            info_list.append({
                "Index": idx,
                "Sector": row['Sector_clean'],
                "Category": row['Category'],
                "Predicted_GI_Status": row['Predicted_GI_Status']
            })
        return info_list
    else:
        return []
```

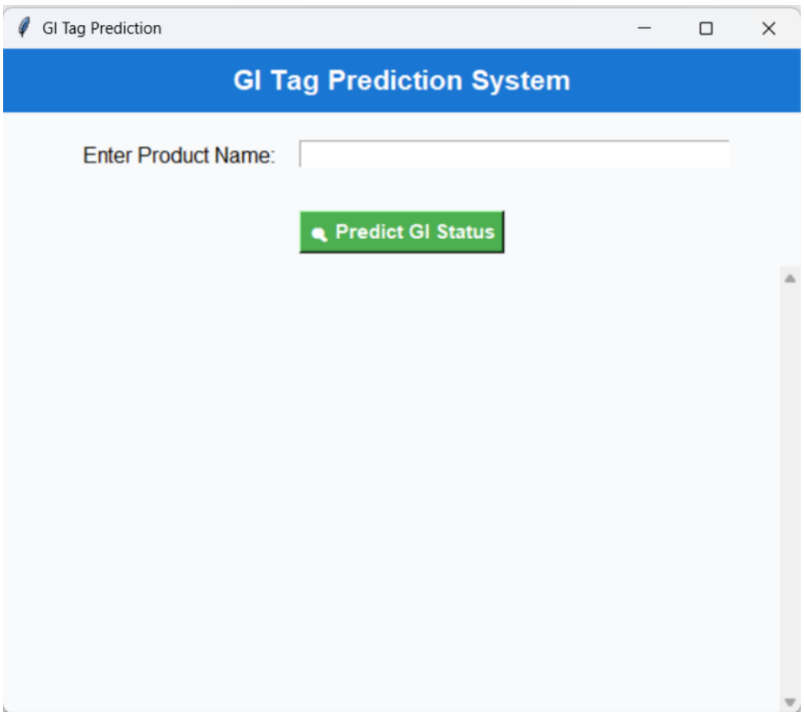
The GUI application enables users to input product name and retrieve corresponding sector, category, and predicted GI status.

```
def on_predict():
    product = entry.get().strip()
    if not product:
        messagebox.showwarning("Input Required", "Please enter a product name.")
        return

    info_list = get_all_rows_info(product)

    if info_list:
        info_text = ""
        for info in info_list:
            info_text += f"Row {info['Index']}: Sector={info['Sector']}, Category={info['Category']},
            Predicted GI Status={info['Predicted_GI_Status']}\n"
    else:
        info_text = "No matching product found in the dataset."

    messagebox.showinfo("Prediction Result", f"Product: {product}\n\n{info_text}")
```

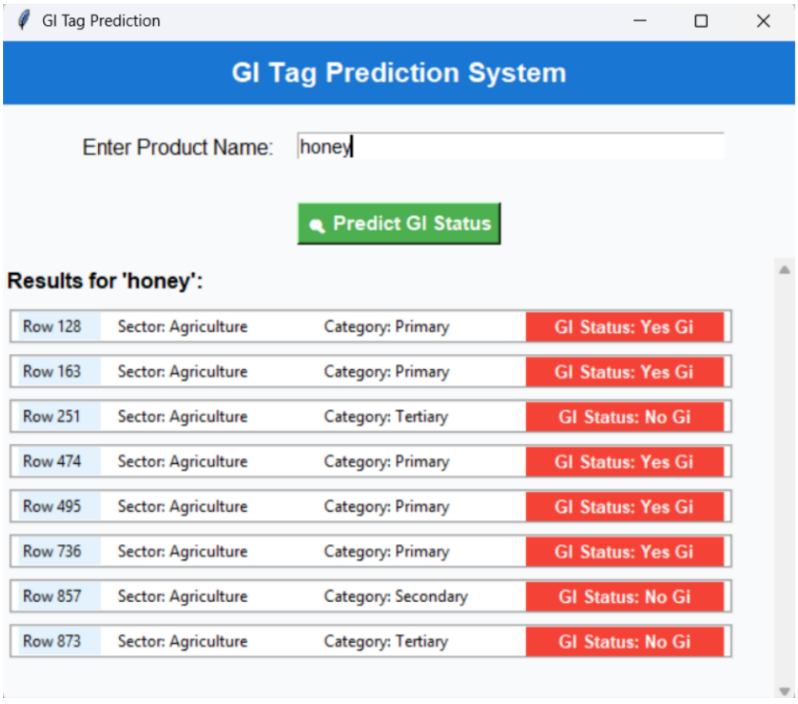
The Tkinter-based GUI provides a user-friendly interface for entering product names and retrieving associated sector, category, and predicted GI status, streamlining access to model insights for non-technical users.

```
root = tk.Tk()
root.title("GI Status ")
root.geometry("500x300")

tk.Label(root, text="Enter Product Name:", font=("Arial", 12)).pack(pady=10)
entry = tk.Entry(root, width=50)
entry.pack(pady=5)

tk.Button(root, text=" Predict GI Status", font=("Arial", 11), command=on_predict).pack(pady=10)

root.mainloop()
```

The application features a Tkinter-based GUI titled 'GI Status Predictor', allowing users to input a product name and receive its predicted GI status.



Upon entering a product name, the GUI retrieves all matching entries from the dataset and displays their sector, category, and predicted GI status in a pop-up window, enabling users to efficiently interpret model outcomes across multiple records



CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 CONCLUSION

The project successfully demonstrates the application of machine learning techniques, specifically logistic regression, to predict the Geographical Indication (GI) status of products listed under India's One District One Product (ODOP) initiative. By analyzing key attributes such as product name, sector, category, and descriptive text, the model identifies patterns that help determine the likelihood of a product being GI-tagged.

The process involved data cleaning, feature engineering, text preprocessing using NLP, and categorical encoding, followed by model training and evaluation. Logistic regression was selected for its interpretability and efficiency in binary classification problems. The evaluation metrics, including accuracy and F1-score, showed that the model performed reasonably well in predicting GI status using available product features.

This study highlights the potential of data-driven approaches in assisting policy formulation, regional branding, and promotion of traditional products. The predictive insights can help government authorities and local entrepreneurs prioritize products for GI registration and marketing strategies, thus contributing to sustainable rural development and cultural preservation.

8.2 FUTURE SCOPE

While the current model provides valuable insights, there remains considerable scope for enhancement and further exploration. There are several potential avenues for extending and enhancing this project in the future.

The first area of improvement lies in the model performance. While logistic regression provided a strong and interpretable baseline, more sophisticated algorithms such as Random Forest, Support Vector Machines (SVM), or Neural Networks can be implemented to achieve higher accuracy and better generalization. Additionally, ensemble learning techniques can be applied to improve predictive stability and robustness.

Another promising direction is the enhancement of text analysis. Incorporating advanced natural language processing (NLP) techniques using deep learning-based models like BERT or Word2Vec can help capture the contextual meaning of product descriptions more effectively. Furthermore, techniques such as sentiment analysis and keyword extraction can be used to understand the regional and cultural significance of products, providing richer insights.

The development of a web-based decision support system can further extend the utility of this project. Such a system can offer an interactive dashboard or web application that predicts the GI likelihood for newly entered products. It can also include data visualization tools that allow policymakers and stakeholders to explore patterns, compare sectors, and analyze regional trends with ease.

Lastly, the project can be integrated into existing policy and administrative frameworks. By linking predictive insights with government portals for ODOP and GI registration, the model can support decision-making processes by helping authorities automatically shortlist products that show a high probability of obtaining GI certification. This integration would streamline workflows, enhance efficiency, and support data-driven policy formulation for regional development and heritage preservation.

REFERENCES

[1] G. T. Kapoor, *"*A Study Of Performance Of One District-One Product Programme (Odop) In Uttar Pradesh*," ResearchGate, vol. 7, no. 28, pp. 216–221.*

[2] Hananto, Pulung & Prananda, Rahandy. (2019). The Urgency Of Geographical Indication As A Legal Protection Instrument Toward Traditionalknowledge In Indonesia. Law Reform. 15. 62. 10.14710/lr.v15i1.23355.

[3]Abd Thalib, "Shortcomings of Geographical Indication in Indonesia: A Critical Appraisal," *Journal of Hunan University Natural Sciences*, vol. 52, no. Volume 52, Issue 5, pp. 63–75, May 2025, doi: https://doi.org/10.55463/issn.1674-2974.52.5.6.

[4]X. F. Quiñones Ruiz *et al.*, "How are food Geographical Indications evolving? – An analysis of EU GI amendments," *British Food Journal*, vol. 120, no. 8, pp. 1876–1887, Aug. 2018, doi: https://doi.org/10.1108/bfj-02-2018-0087.

[5] An Introduction to Logistic Regression Analysis and Reporting," *ResearchGate*. https://www.researchgate.net/publication/242579096_An_Introduction_to_Logistic_Regression_Analysis_and_Reporting.

[6] H. A. Salman, A. Kalakech, and A. Steiti, "Random Forest Algorithm Overview," *Babylonian Journal of Machine Learning*, vol. 2024, pp. 69–79, Jun. 2024, doi: https://doi.org/10.58496/bjml/2024/007.

[7] M. Suyal and P. Goyal, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 43–48, Jul. 2022, doi: https://doi.org/10.14445/22315381/ijett-v70i7p205.

[8] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A Comparative Analysis of XGBoost," *ResearchGate*, Nov. 2019, doi: https://doi.org/10.48550/arXiv.1911.01914.

[9] M. H. Almaspoor, A. Safaei, A. Salajegheh, and B. Minaei-Bidgoli, "Support Vector Machines in Big Data Classification: A Systematic Literature Review," Aug. 2021, doi: https://doi.org/10.21203/rs.3.rs-663359/v1.

[10] GeeksforGeeks, "Univariate, Bivariate and Multivariate data and its analysis," GeeksforGeeks, Jul. 11, 2025. https://www.geeksforgeeks.org/data-analysis/univariate-bivariate-and-multivariate-data-and-its-analysis/

[11]GeeksforGeeks, "Graphical User Interface," GeeksforGeeks, Jul. 26, 2025. https://www.geeksforgeeks.org/computer-graphics/what-is-graphical-user-interface/