MALLA REDDY
ENGINEERING COLLEGE AND MANAGEMENT SCIENCES
(AN UGC AUTONOMOUS INSTITUTION)
Approved by AICTE , Accredited by NBA (CSE, IT, EEE, ECE)
Kistapur Village, Mediciti Road, Medchal, Hyderabad – 501401, Telangana, India

MACHINE LEARNING MODEL FOR FOOD ADULTERATION ANALYSIS

PROJECT GUIDE
Dr. V. MANISARMA. Ph.D, Professor
DEPARTMENT OF CSM

PRESENTED BY: 4th Year CSD BATCH 2
RAMPOLA BHARTH REDDY (21UJ1A6731)
VEMUGANTI SUSHANTH (21UJ1A6739)
NIRUGONDA SUSHMA (22UJ5A6709)
CHINNI SAI KRISHNA (22UJ5A6706)

INDEX

ABSTRACT

Food Adulteration is a process by which quality, or the nature of given substance is reduced through the addition of foreign substance which gives flavor to the food but dangerous to the human health. Adulteration consists of large number of practices such as mixing substitutions, concealing the quality etc.Our research aims to develop a machine learning model by analyzing the rate of adulterant used in a food product and predict the action according to the rate of adulterant used within less time and with good efficiency and also aims to achieve high accuracy in detection, providing a tool for stakeholders in the food industry and to the food safety organizations.

INTRODUCTION

- Along the food supply chain, food products may become contaminated with various types of safety hazards,
- biological hazards (bacteria, viruses, and parasites), chemical hazards (heavy metals, pesticides, and mycotoxins), or physical hazards (metal fragments and pieces of glass).
- The presence of these hazards can affect the safety of food or feed products, with potentially detrimental effects on human and animal health.
- Monitoring potential food safety hazards along the entire food supply chain is important in order to guarantee the correct functioning of food safety management systems.

MOTIVATION

- Our motive of this research is to prevent from getting health issues by banning highly adulterated food products by detecting the rate of adulterant used in a product and this can help the organizations .
- Food adulteration poses significant health risks to consumers. Contaminated or adulterated food products can lead to serious health issues, including poisoning, allergies, and long-term diseases.
- Primary motivation is the need for a cost-effective and less time consumption to verify whether the food product is very dangerous.

PROBLEM STATEMENT

- Human health and consumer safety are seriously at risk by food adulteration, and traditional methods of detection are expensive, time-consuming, and usually poor for complicated or new adulterants.
- New models are required for detecting food product adulteration. By analysing big, complex datasets for pattern identification and immediate recognition.
- Machine learning provides problems including model assumptions, kinds of food change, and irregular data have to be overcome. In order to improve food safety and quality tracking, this work proposes to develop an ML model for accurate, adjustable food contamination detection.
- The model should be able to identify common adulterants in foods like milk, spices, and oils and it gives what target action should take on a particular adulterant rate

LITERATURE SURVEY

S.NO	AUTHOR	YEAR	TITLE	KEY CONCEPT
1	K.Goyal, P.Kumar & K.Verma	2022	Food Adulteration Detection using Artificial Intelligence.	In this paper, we came to know the role of Artificial Intelligence in food adulteration detection.
2	Changquan Huang	2022	A Machine Learning Method for the Quantitative Detection of Adulterated Meat	Meat adulteration causes harmful diseases like allergies, rashes etc. This paper thought us to know detailed view of adulterants used in meet and identification of those through ML model.
3	Zaukuu, JL.Z., Adam, M.N., Nkansah, A.A.	2024	Detection and quantification of groundnut oil adulteration with machine learning.	This research aimed to apply machine learning model to detect adulterants used in groundnut oil.
4	Esmael Ahmed	2024	Detection of honey adulteration using machine learning.	The study uses hyperspectral imaging, a promising tool for food quality assurance, to classify and predict adulteration in honey.
5	Junming Han, Tong Li	2022	Using Machine Learning Approaches for Food Quality Detection	This paper shows identifying the freshness of fruits through experimental results and discusses the overfitting of machine learning based on the experimental results.

RESEARCH GAPS

- There is a shortage of large-scale datasets for food adulteration, it is difficult to find the dataset.
- Many models require high-cost infrastructure, posing challenges for broad implementation, especially in developing regions.
- Existing machine learning models have trouble detecting newly created or unknown adulterants that was not included in their training set.
- When new adulteration methods created, existing systems are unable to automatically update and adapt their detection processes.
- ML models usually fail to take into consideration the various ways that food adulteration practices and trends change among regional marketplaces and locations.
- Costs and return on investment of deploying based on machine learning adulteration detection systems at volume was not fully investigated.

PROPOSED SYSTEM

- The approach aims to develop a model that can automatically identify and classify food adulterants across a wide range of food products with higher speed, accuracy, and scalability.
- The system works by utilizing a dataset that contains both pure and adulterated samples of various food items. Key features of each sample—such as color, texture, chemical composition, and physical properties—are extracted and analyzed through advanced feature extraction techniques. These features serve as the input for the machine learning model, enabling it to differentiate between pure and adulterated samples with high precision.and the action taken by the adulterant

APPLICATIONS

- Restaurants.
- Hostels.
- Food Industries.
- Quality Control in Food Processing.
- Consumer Mobile Applications.
- Supply Chain Monitoring

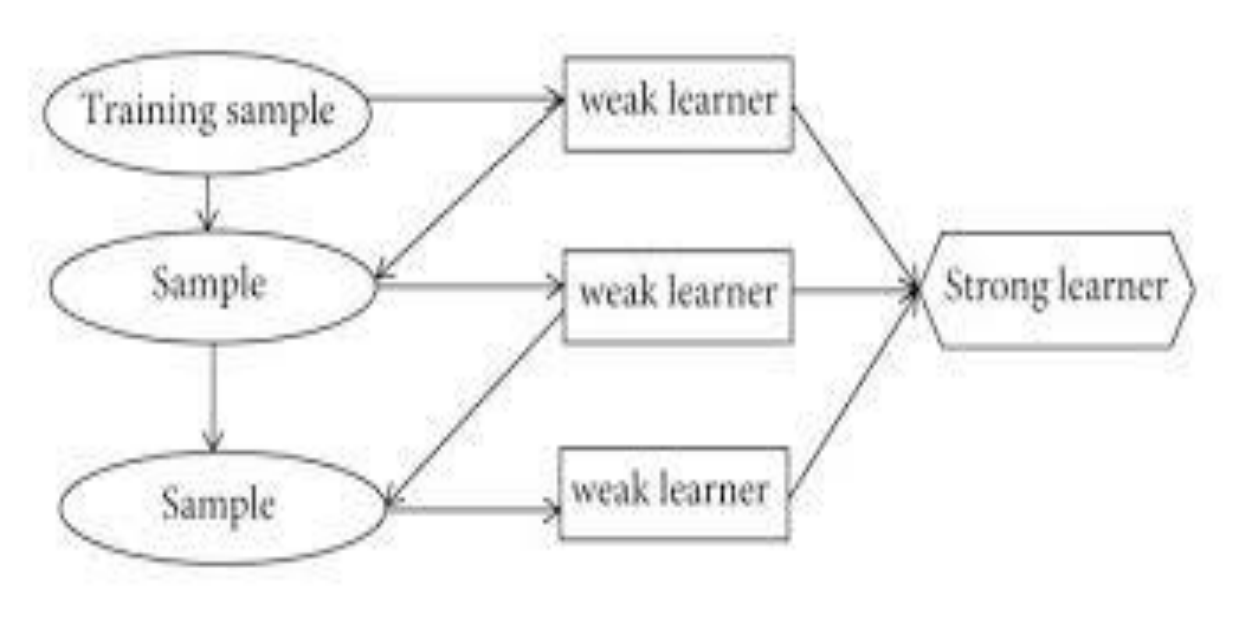
ADVANTAGES

- Accuracy
- Import food safety
- Cost effective
- Real time monitoring
- Economic benefits

EXISTING METHODOLOGY:

Gradient Boost Classifier

- In existing methodology ,Gradient Boost Classifier machine learning model is used to take the actions on highly adulterated food product.
- In this the performance evaluation used like accuracy ,precision recall,f1 score etc.



DRAWBACKS

It takes more training time.

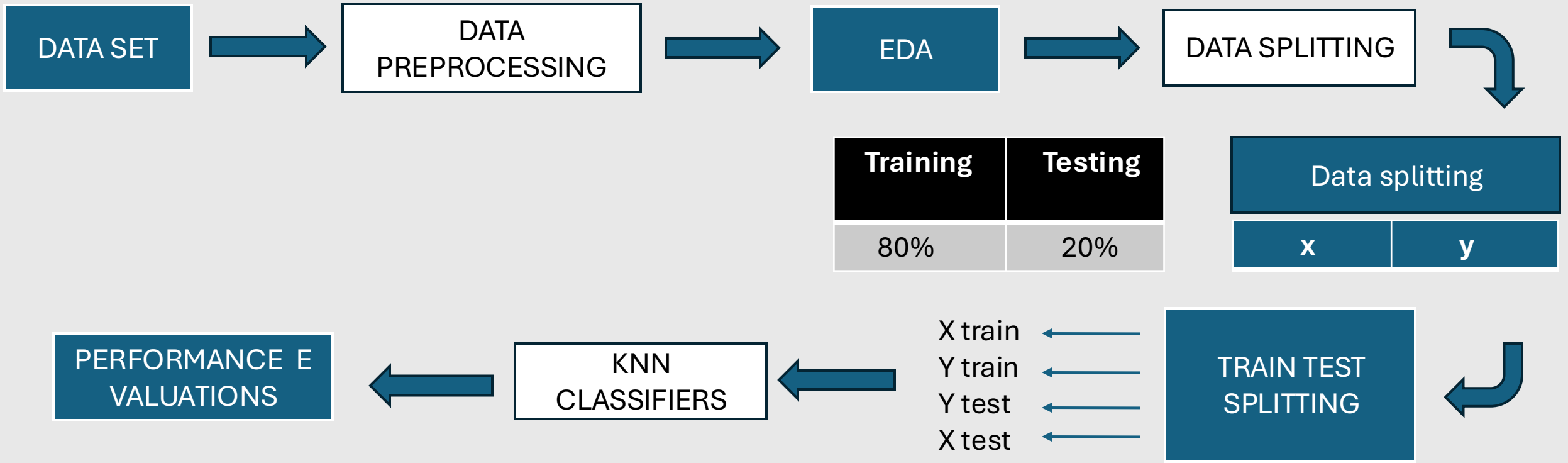
It requires more memory space where it contains duplicate and null values as well.

It does not provide accurate actions to be taken on food products.

Retraining could quickly become very expensive

PROPOSED METHODOLOGY

KNN classifier:



DATA SETS

- NULL
- DUPLICATE
- LABEL ENCODING
- SMOTE
- RESAMPLING
- STANDARD SCALING

DATA PREPROCESSING

- NUMPY
- PANDAS

EDA

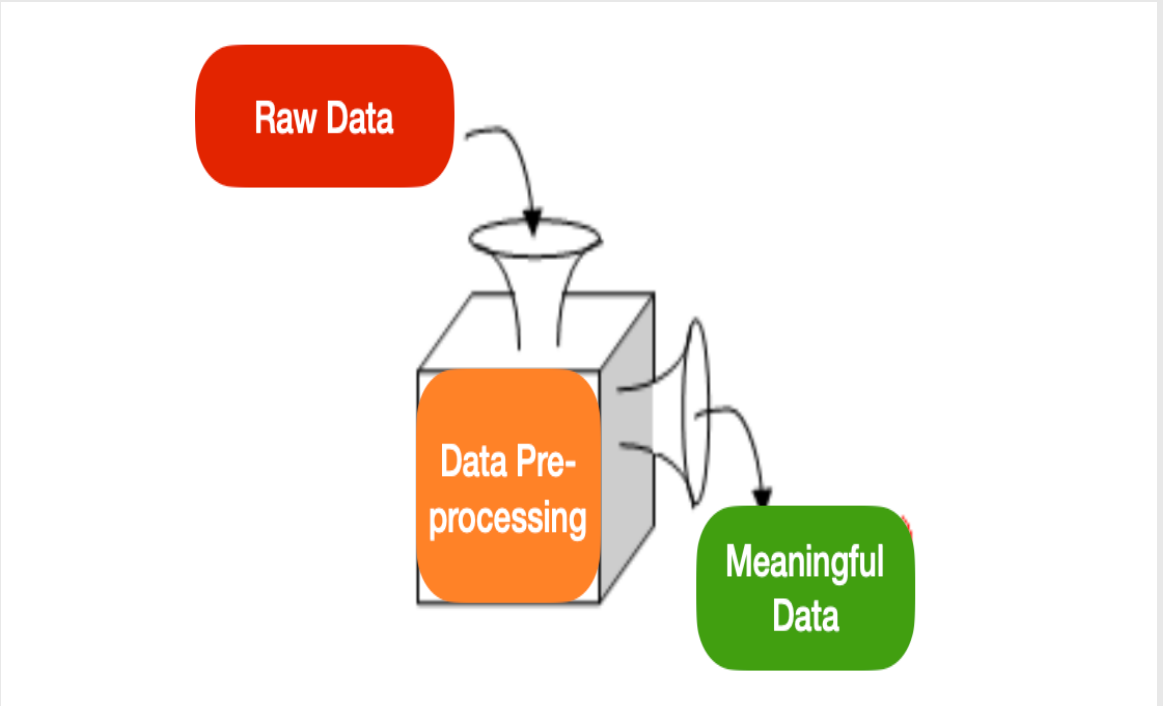
- MATPLOTLIB
- SEABORN
- COUNTPLOT

PREPROCESSING

Data Preprocessing is nothing but cleaning data, transforming data, splitting data. In preprocessing we use NumPy and panda's library.

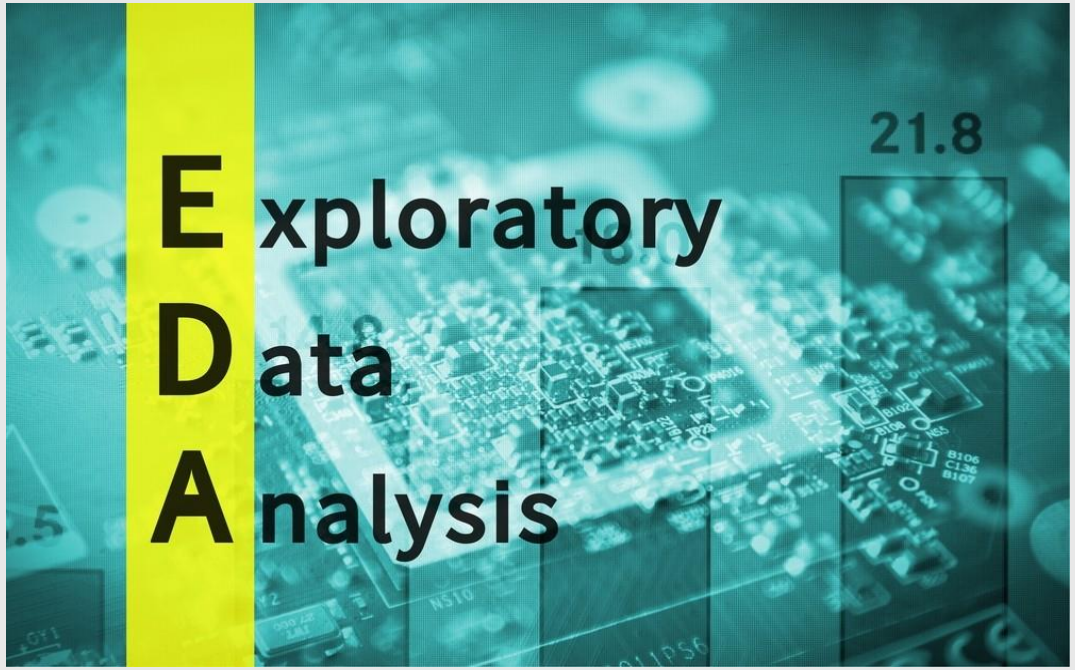
Steps in preprocessing: -

- **Handling Null Values:** -Handling null values (or missing data) is a critical step in data preprocessing. null values occur during data entry errors, incomplete data.
- **Handling duplicate data:** - Duplicate data refers to records that are same or very similar within a dataset. This will occur due to: data entry errors, merging datasets, data collections issues.
- **Label encoding**: - it is a technique used to convert categorical variables into numerical format be used in machine learning algorithms. Many algorithms require numerical input and do not work directly with categorical data.



EDA (exploratory data analysis)

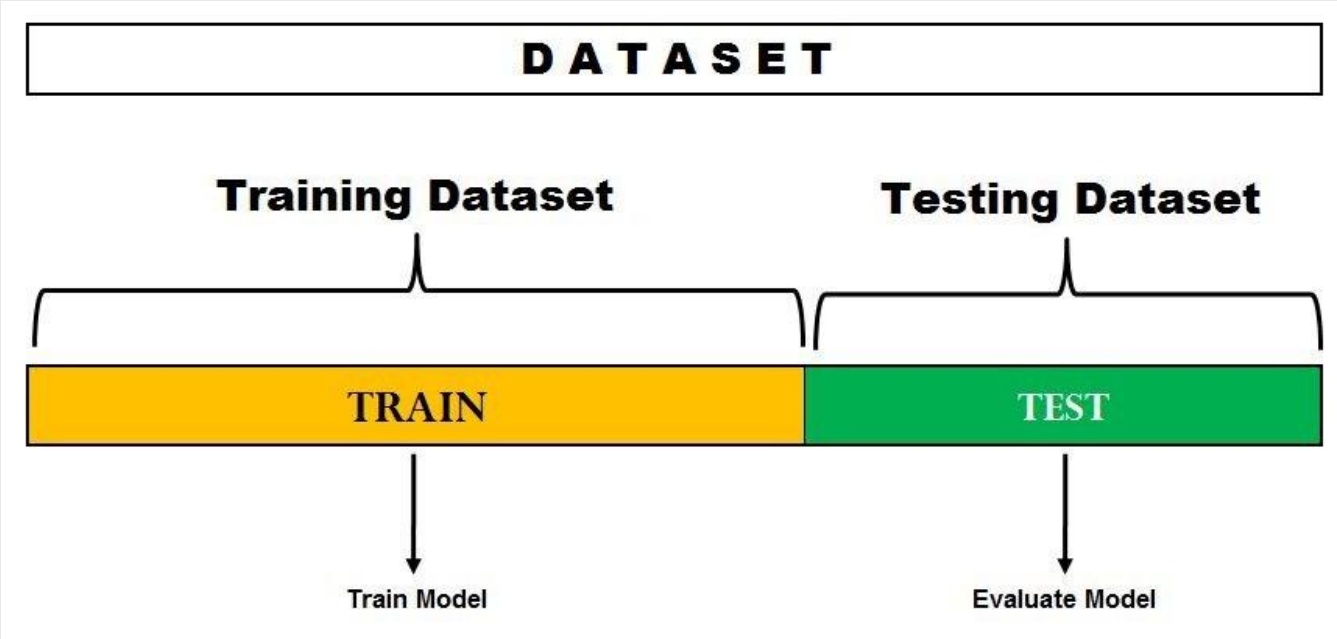
- Exploratory Data Analysis (EDA) is an important step in analyzing data. It helps us understand the main features of a dataset by examining and visualizing it. The main goals of EDA are to learn about the data, find patterns, identify unusual data points, create hypotheses, and prepare for modeling.
- The EDA process starts with collecting the data from various sources. Next, we clean the data by handling missing values, removing duplicates, and ensuring that each column has the correct data type. After cleaning, we calculate descriptive statistics, which provide insights into the average, middle, and spread of the data.



DATA SPLITING

Data splitting is a way to divide your dataset into different parts so you can train and test a machine learning model effectively. Imagine you have a big bag of candies, and you want to see how good you are at guessing which candies are your favorites. To do this, you need to divide the candies into two groups:

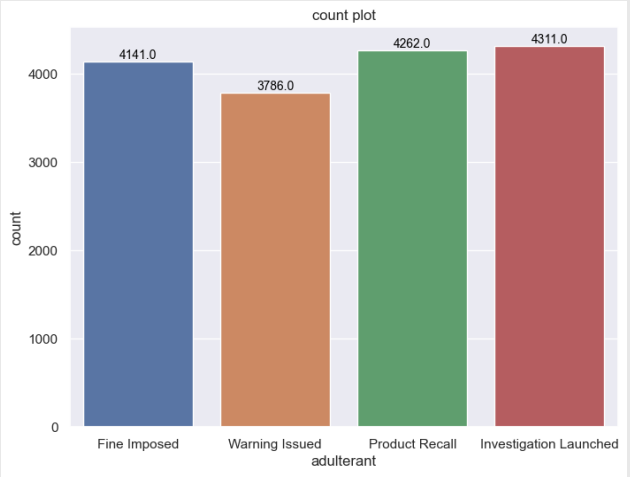
- Training Group**:** This is the group of candies you will use to practice. You taste these candies and learn which ones you like the most. This helps you understand the different flavors and types.
- Testing Group: This is the group of candies you set aside. After you've practiced and learned, you use these candies to test your guessing skills. You try to guess which ones are your favorites without tasting them first.



SMOTE

- SMOTE stands for Synthetic Minority Over-sampling Technique.
- Smote is the one of the most commonly used over sampling method to solve the imbalance problem.
- It aims to balance class distribution by randomly increasing the minority class examples by replicating them.
- Smote synthesizes new minority instances between the minority instances .

Before smote

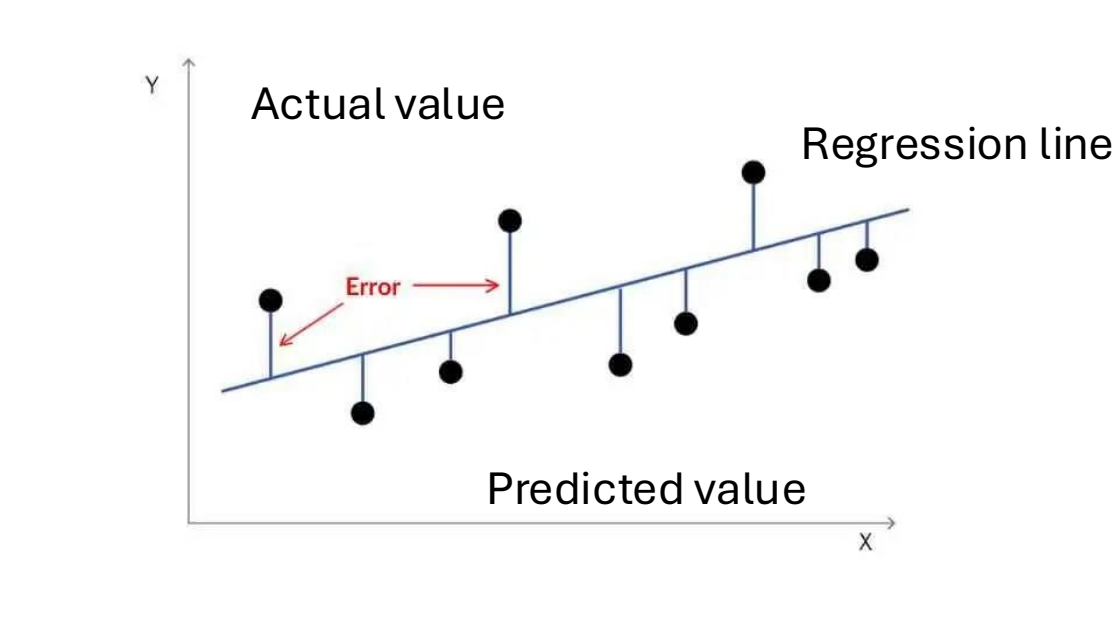


After smote

GRADIENT BOOST ALGORITHM

- Optimizing the loss function

  Loss= [actual – predicted]
- The trees are bigger in gradient boost
- The leaf nodes are 8 to 32



Steps to follow..

ACTUAL	PREDICTED	RESIDUALS	RESIDUAL PREDICTED
174	171	3	3.5
176	171	5	4.2
169	171	-2	-1

- Compute the average of target column , then the predicted value will come
- Residual =[actual –predicted]
- Fit a model on residual that model will predict the residual

  Using RM1(residual model 1)

  (Predicted + learning rate(LR) + residual predicted)..
- Update the default residual
- Final prediction=base value +(LR * 1st residual predict by RM1)+(LR * 2nd residual predict by RM2)….

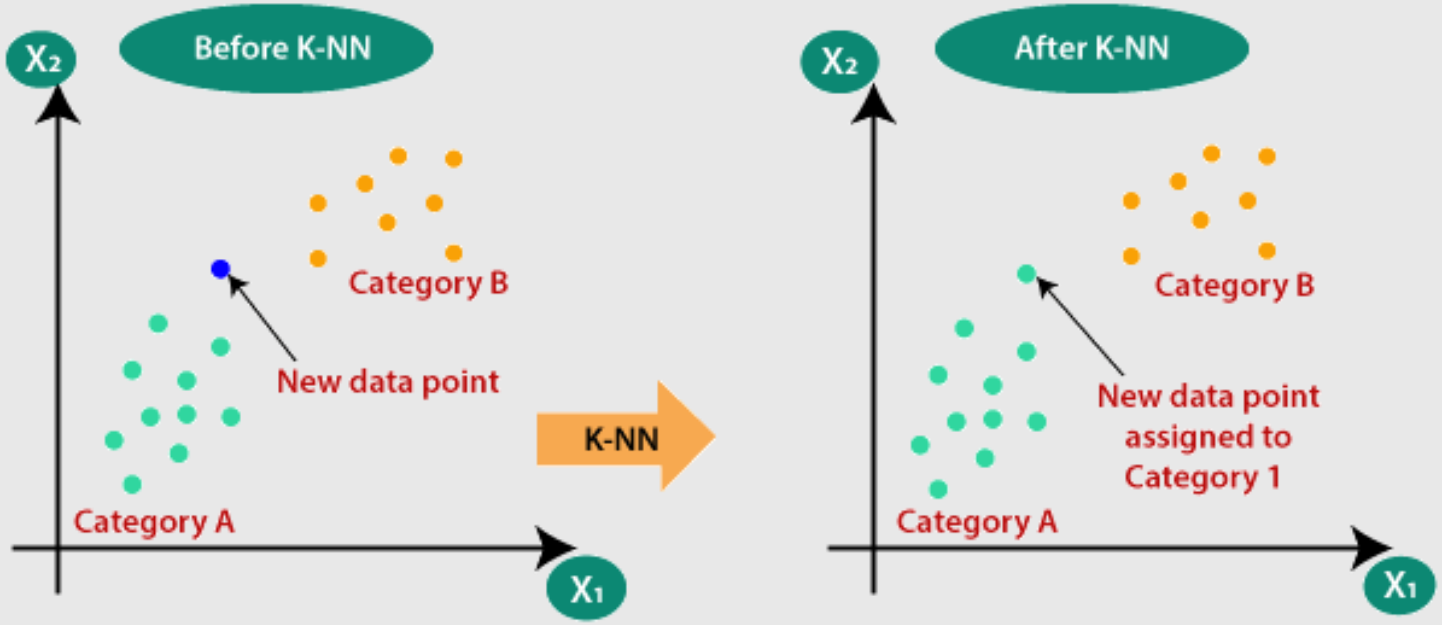
DRAWBACKS

- **Overfitting**: The model can learn too much from the training data and not do well on new data. It’s like memorizing answers instead of understanding them.
- **Slow Training**: It can take a long time to build the model, especially if there is a lot of data. This can be a problem if you need quick answers.
- **Sensitive to Mistakes**: If the training data has a lot of mistakes or unusual values, the model can get confused and make wrong predictions.
- **Hard to Adjust**: To get the best results, you often need to change several settings. This can be tricky and take a lot of time, especially for beginners.
- **Difficult to Understand**: While it can show which parts of the data are important, the whole model can be hard to explain. This makes it tough to say why it made a certain choice.
- **Uses a Lot of Memory**: It can need a lot of computer memory, especially with large datasets, which might be a problem on some computers.

KNN (K-NEAREST NEIGHBOR)

KNN (K-NEAREST NEIGHBOR)

- KNN is a lazy algorithm
- First it stored the training data .when we send the testing data then the training is starting.



ADVANTAGES

- **Simple to Understand**: KNN is easy to understand.
- **No Training Time**: KNN doesn’t need a long training phase. It just remembers all the data, use it right away.
- **Works with Different Data**: KNN handles different kinds of data, like numbers and categories, which makes it flexible.
- **Good for Small Datasets**: KNN performs well when you have a small amount of data. It make accurate predictions with fewer points.
- **Classify Many Groups**: KNN sorts data into more than two categories easily, which is helpful for various problems.
- **Effective with Clear Patterns**: If the data has clear patterns, KNN will be very accurate at guessing the right answers

SOURCE CODE

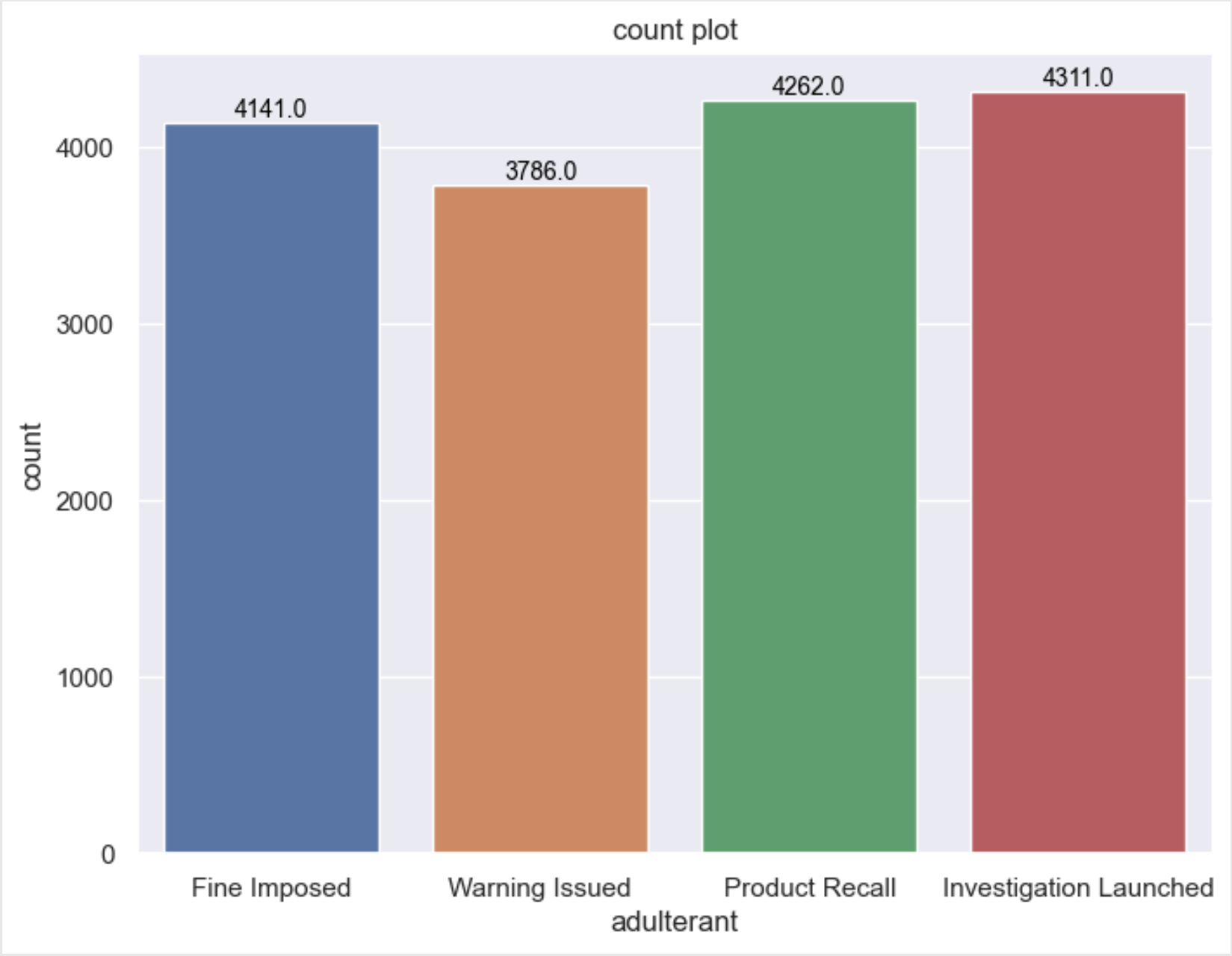
Data preprocessing

Data cleaning

```
      # to check the null values:
       df.isnull().sum()
     # to check the duplicate values:
        df.duplicated().sum()
        df = resample(df, replace=True, n_samples=16500,
random_state=42)
```

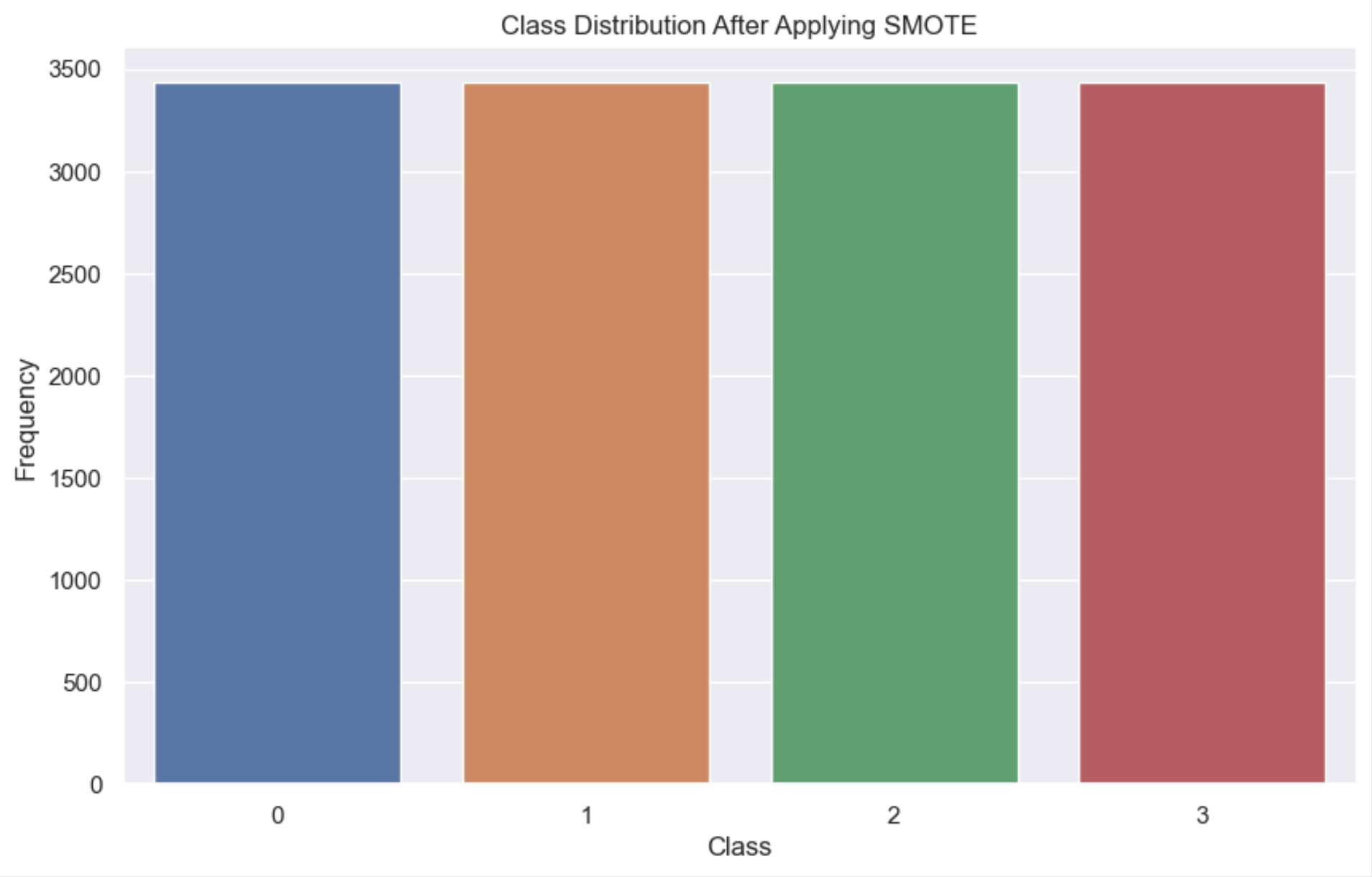
Exploratory data analysis

```
# count plot for adulterant column:
sns.set(style="darkgrid")
plt.figure(figsize=(8,6))
ax=sns.countplot(x='action_taken',data=df)
plt.title('count plot')
plt.xlabel('adulterant')
plt.ylabel('count')
for p in ax.patches:
ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
ha='center', va='center', fontsize=10, color='black', xytext=(0, 5),
textcoords='offset points')
plt.show()
```


count plot
4141.0
3786.0
4262.0
4311.0
4000
3000
2000
1000
0
count
Fine Imposed
Warning Issued
Product Recall
Investigation Launched
adulterant

Applying smote

```
# Visualize the class distribution
plt.figure(figsize=(10, 6))
sns.barplot(x=class_counts.index, y=class_counts.values)
plt.xlabel('Class')
plt.ylabel('Frequency')
plt.title('Class Distribution After Applying SMOTE')
plt.show()
```


Class Distribution After Applying SMOTE
3500
3000
2500
2000
1500
1000
500
0
Frequency
0
1
2
3
Class

Gradient Boosting Classifier

```
#gbc model:
if os.path.exists('GradientBoostClassifier.pkl'):
    #load the model:
    gbc=joblib.load('GradientBoostClassifier.pkl')
    print('model loaded successfully')
    predict=gbc.predict(x_test)
    calculateMetrics("KNeighborsClassifier",predict,y_test)
else:
    #train the model:
    gbc = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
    gbc.fit(x_resampled,y_resampled)
    #saving model:
    joblib.dump(gbc,'GradientBoostClassifier.pkl')
    print('model saved successfully')
    predict=gbc.predict(x_test)
    calculateMetrics("GradientBoostClassifier",predict,y_test)
```

Output

- model loaded successfully
  Accuracy : 77.6969696969697
  Precision : 77.74917243007128
  Recall   : 78.3456876406422
  FSCORE   : 77.8447322258609

KNN classifier

```
#knn model:
if os. Path. exists('K Neighbors Classifier.pkl'):
    #load the model:
    Knn =joblib.load('KNeighborsClassifier.pkl')
    print('model loaded successfully')
    Predict=knn.predict(x_test)
    calculateMetrics("KNeighborsClassifier",predict,y_test)
else:
    #train the model:
    knn=KNeighborsClassifier()
    knn.fit(x_resampled,y_resampled)
    #saving model:
    joblib.dump(knn,'KNeighborsClassifier.pkl')
    print('model saved successfully')
    predict=knn.predict(x_test)
    calculateMetrics("KNeighborsClassifier",predict,y_test)
```

Output

- model loaded successfully
- K Neighbors Classifier Accuracy : 100.0
- K Neighbors Classifier Precision : 100.0
- K Neighbors Classifier Recall   : 100.0
- K Neighbors Classifier FSCORE   : 100.0

RESULT ANALYSIS

COMPARISION MATRIX:

Matrix	Gradient boost classifier	KNN Classifier
Accuracy	77.69%	100%
Precision	77.74%	100%
Recall	78.34%	100%
F1 Score	77.84%	100%

SOFTWARE DETAILS:

We also used some of the libraries. They are:

- NUMPY
- Pandas
- matplotlib
- seaborn

Software we have used in our research is:

- Python 3.7.6,Jupyter notebook.

REFERENCES

- K.Goyal, P.Kumar, Esmael Ahmed, Detection of honey adulteration using machine learning, (2024). (https://doi.org/10.1371/journal.pdig.0000536)
- Changquan Huang, A Machine Learning Method for the Quantitative Detection of Adulterated Meat Feb 20 (2022). https://doi.org/10.3390/foods11040602
- Zaukuu, JL.Z., Adam, M.N., Nkansah, A.A. et al. Detection and quantification of groundnut oil 14, 20931 (2024) https://doi.org/10.1038/s41598-024-70297-7
- Esmael Ahmed, Detection of honey adulteration using machine learning, (2024). https://doi.org/10.1371/journal.pdig.0000536
- Junming Han, Tong Li, Using Machine Learning Approaches for Food Quality Detection (2022). https://doi.org/10.1155/2022/6852022

conclusion

- Supervised learning is a powerful tool for classifying food products. By using labelled data of food with pure and adulterated rate, machine learning models. This technology not only reduces adulteration levels in food but also promotes sustainable practices in agriculture.
- In conclusion, for food adulteration classification, we used two machine learning models: the Gradient Boost Classifier and KNN Classifier. Each model's performance was measured based on four metrics: Accuracy, Precision, Recall, and F1-Score.
- The existing method GBC model achieved an accuracy of 77.69%, with a precision of 77.74%, recall of 78.34%, and F1-Score of 77.84%.
- The proposed method KNN Classifier performed better, with an accuracy of 100%, precision of 100%, recall of 100%, and F1-Score of 100%.
- KNN Classifier overtook Gradient Boost Classifier across all metrics, suggesting it may be a more reliable choice for accurately identifying actions on particular food product.

THANK YOU