TRAINING REPORT

OF

SUMMER TRAINING, UNDERTAKEN AT

NATIONAL INSTITUTE OF ELECTRONICS & INFORMATION TECHNOLOGY(NIELIT)

ON

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

PROJECT: MILK QUALITY PREDICTION

SUBMITTED BY:

Mr. JASHANDEEP SINGH

Mr. LAGANPREET SINGH

SUBMITTED TO:

Ms. ANITA BUDHIRAJA

Joint Director, NIELIT

Dr. SARWAN SINGH

Joint Director, NIELIT

ABSTRACT

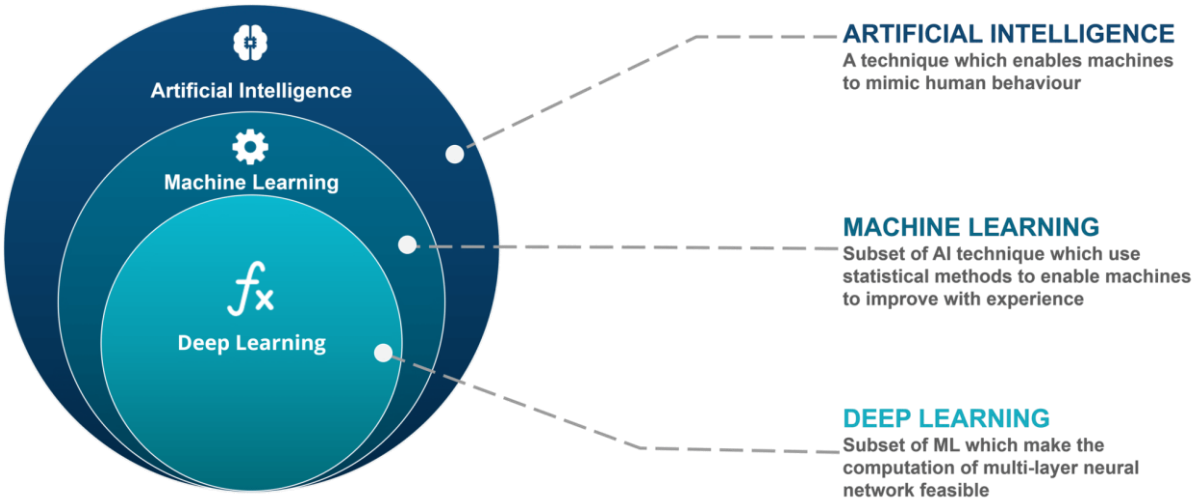
Milk is the main dietary supply for every individual. High-quality milk shouldn't contain any adulterants. Dairy products are sold everywhere in society. Yet, the local milk vendors use a wide range of adulterants in their products, permanently altering the evaporated. Using milk that has gone bad can have serious health consequences. On October 18 of this year, the Food Safety and Standards Authority of India (FSSAI), the nation's top food safety authority, released the final result of the National Milk Safety and Quality Survey (NMSQS) and declared the milk readily available in India to be "mostly safe." According to an FSSAI survey, 68.4% of the milk in India is tainted. The quality of milk cannot be checked by any equipment or special system. Milk that has not been pasteurized has not been treated to get rid of harmful bacteria. Infected raw milk may contain Salmonella, Campylobacter, Cryptosporidium, E. coli, Listeria, Brucella, and other dangerous pathogens. These microorganisms pose a major risk to your family's health. Manually analyzing the various milk constituents can be very challenging when determining the quality of the milk. Analyzing and discovering with the aid of machine learning can help with this endeavor. Here a machine learning-based milk quality prediction system is developed. The proposed technology has shown 99.99% classification accuracy.

Table of content

CHAPTER 1 – INTRODUCTION

1.1 Artificial Intelligence and Machine learning

Artificial Intelligence and Machine Learning are two rapidly evolving fields at the forefront of technological advancements. They both revolve around the idea of creating intelligent systems capable of performing tasks that traditionally require human intelligence. While AI is a broader concept, encompassing a wide range of techniques and applications, Machine Learning is a subset of AI that focuses on creating algorithms and models that enable machines to learn and improve from experience.



1. **Artificial Intelligence (AI):**

Artificial Intelligence is the theory and development of computer systems capable of performing tasks that typically require human intelligence. These tasks can include reasoning, problem-solving, perception, learning, language understanding, and decision making. The ultimate goal of AI is to create machines that can mimic, simulate, or even surpass human intelligence in various domains. AI systems can be broadly categorized into two types:

a. **Narrow AI (Weak AI):** This type of AI is designed to perform specific tasks or solve particular problems. Examples include virtual personal assistants like Siri, speech recognition systems, and recommendation algorithms used by online platforms.

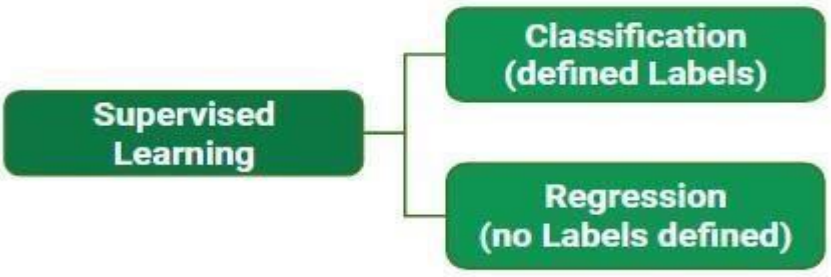
b. **General AI (Strong AI):** General AI aims to possess human-like intelligence, enabling machines to understand, learn, and perform any intellectual task that a human being can do.

Achieving this level of AI remains a significant challenge and is still largely theoretical.

2. **Machine Learning (ML):**

Machine Learning is a subset of AI that focuses on developing algorithms and statistical models to enable computers to learn and improve from experience automatically. Instead of being explicitly programmed, these algorithms learn patterns from data and make predictions or decisions based on that learning. There are three main types:

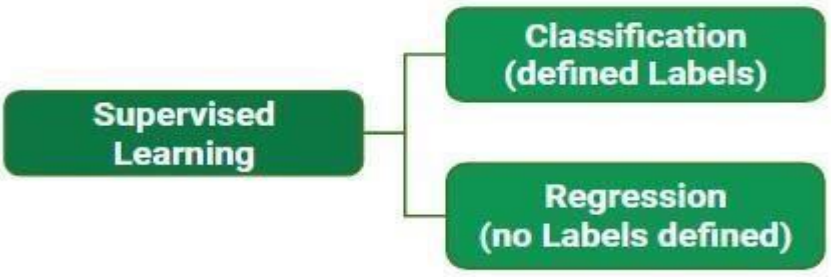
a. **Supervised Learning**: In this approach, the algorithm is trained on labelled data, where the correct output is provided. The model learns to map inputs to outputs, making it capable of making predictions on new, unseen data.



b. **Unsupervised Learning**: Unsupervised learning deals with unlabelled data, where the algorithm tries to find patterns and relationships within the data without specific guidance.

Clustering and dimensionality reduction are common tasks in unsupervised learning.

a. **Supervised Learning**: In this approach, the algorithm is trained on labelled data, where the correct output is provided. The model learns to map inputs to outputs, making it capable of making predictions on new, unseen data.



b. **Unsupervised Learning**: Unsupervised learning deals with unlabelled data, where the algorithm tries to find patterns and relationships within the data without specific guidance.

Clustering and dimensionality reduction are common tasks in unsupervised learning.

1. **Natural Language Processing (NLP):**

NLP enables machines to understand, interpret, and generate human language. Applications include sentiment analysis, chatbots, language translation, speech recognition, and text summarization.

2. **Recommender Systems:**

Recommender systems use Machine Learning to suggest products, movies, music, or content based on a user's preferences and behaviour. Examples include personalized recommendations on streaming platforms and e-commerce product recommendations.

3. **Autonomous Vehicles**:

Machine Learning plays a crucial role in developing self-driving cars. It helps in object detection, lane detection, real-time decision-making, and mapping for navigation.

4. **Healthcare**:

Machine Learning is used for medical image analysis, disease diagnosis, drug discovery, personalized treatment plans, and predicting patient outcomes.

5. **Financial Services**:

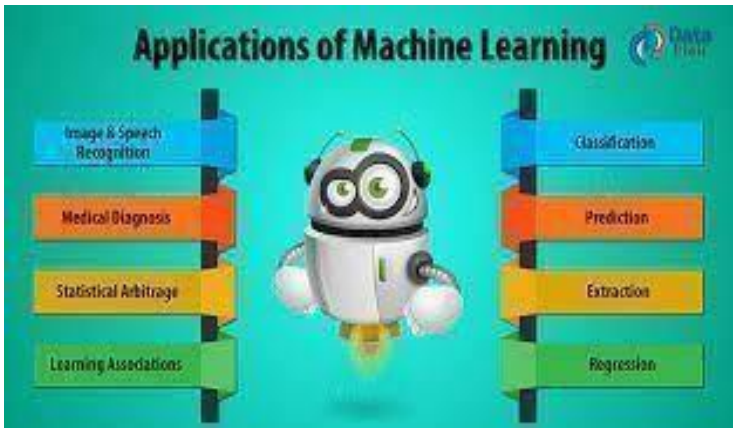
In finance, Machine Learning is applied for credit scoring, fraud detection, algorithmic trading, risk assessment, and customer service chatbots.

6. **Predictive Maintenance**:

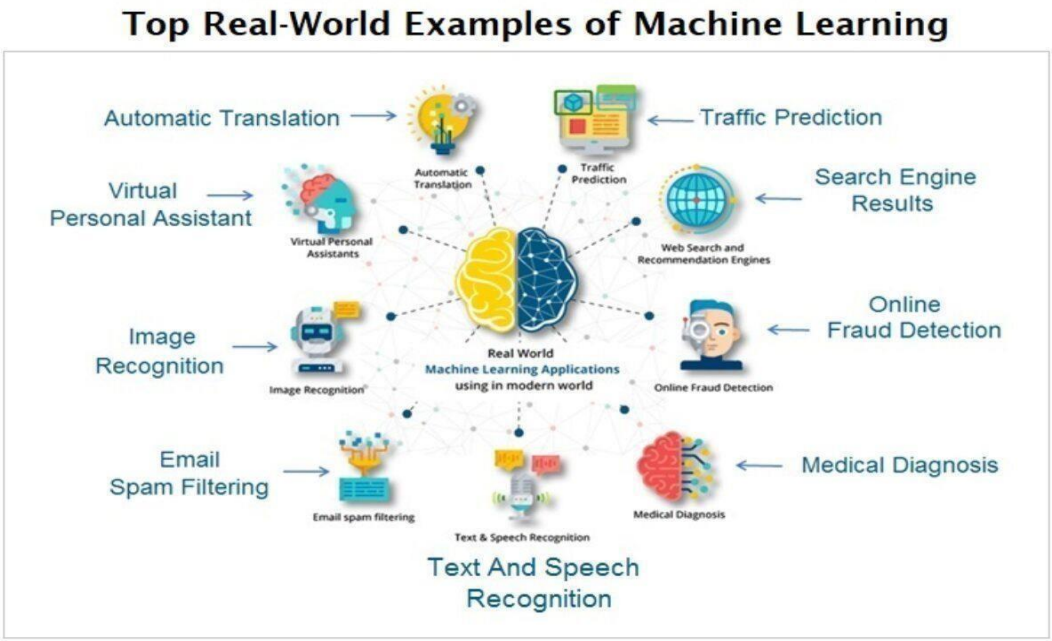
In manufacturing and industrial sectors, Machine Learning is used for predictive maintenance, optimizing equipment maintenance schedules, and reducing downtime.

7. **Fraud Detection**:

Machine Learning models can detect fraudulent transactions, account takeovers, and identify unusual patterns in financial activities.



1.3 Real World Examples of Machine Learning



1. Speech & Image Recognition

Computer Speech Recognition *or* Automatic Speech Recognition helps to convert speech into text. Many applications convert the live speech into an audio file format and later convert it into a text file.

Similar to speech recognition, Image recognition is also the most widely used example of Machine Learning technology that helps identify any object in the form of a digital image.

2. Traffic alerts using Google Map

Google map uses different technologies, including machine learning which collects information from different users, analyse that information, update the information, and make predictions.

With the help of predictions, it can also tell us the traffic before we start our journey.

Machine Learning also helps identify the best and fastest route while we are in traffic using

Google Maps

3. Chatbot (Online Customer Support)

A chatbot is the most widely used software in every industry like banking, Medical, education, health, etc. You can see chatbots in any banking application for quick online support to customers. These chatbots also work on the concepts of Machine Learning. The programmers feed some basic questions and answers based on the frequently asked queries. So, whenever a customer asks a query, the chatbot recognizes the question's keywords from a database and then provides appropriate resolution to the customer. This helps to make quick and fast customer service facilities to customers.

4. Google Translation

Suppose you work on an international banking project like French, German, etc., but you only know English. In that case, this will be a very panic moment for you because you can't proceed further without reviewing documents. Google Translator software helps to translate any language into the desired language. So, in this way, you can convert French, German, etc., into English, Hindi, or any other language. This makes the job of different sectors very easy as a user can work on any country's project hassle-free.

5. Prediction

Prediction system also uses Machine learning algorithms for making predictions. There are various sectors where predictions are used. For example, in bank loan systems, error probability

can be determined using predictions with machine learning. For this, the available data are classified into different groups with the set of rules provided by analysts, and once the classification is done, the error probability is predicted.

6. Extraction

One of the best examples of machine learning is the extraction of information. In this process, structured data is extracted from unstructured data, and which is used in predictive analytics tools. The data is usually found in a raw or unstructured form that is not useful, and to make it useful, the extraction process is used.

7. Self-driving cars

The future of the automobile industry is self-driving cars. These are driverless cars, which are based on concepts of deep learning and machine learning. Some commonly used machine learning algorithms in self-driving cars are Scale-invariant feature transform (SIFT), AdaBoost, TextonBoost, YOLO (You only look once).

8. Ads Recommendation

Nowadays, most people spend multiple hours on google or the internet surfing. And while working on any webpage or website, they get multiples ads on each page. But these ads are different for each user even when two users are using the same internet and on the same location. These ads recommendations are done with the help of machine learning algorithms. These ads recommendations are based on the search history of each user. For example, if one user searches for the Shirt on Amazon or any other e-commerce website, he will get start ads recommendation of shirts after some time.

1.4 Python Programming Language

Python Language Introduction Python is a widely used general-purpose, high level programming language. It was initially designed by **Guido Van Rossum in 1991** and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

1.5 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Small Talk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

1.6 Features of Python

Python's features include −

Easy-to-learn – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read − Python code is more clearly defined and visible to the eyes.

Easy-to-maintain − Python's source code is fairly easy-to-maintain.

A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases − Python provides interfaces to all major commercial databases.

GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

It supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

IT supports automatic garbage collection.

1.7 Role of Python in Machine Learning:

Python plays a significant role in the field of machine learning and is widely regarded as one of the most popular and versatile programming languages for this domain. Its role in machine learning can be attributed to several key factors:

1. **Rich Ecosystem of Libraries:** Python offers an extensive collection of libraries and frameworks specifically designed for machine learning and data science. Some of the most prominent ones include:

- NumPy: A fundamental library for numerical computation, which forms the backbone of many other machine learning libraries.

- pandas: A powerful library for data manipulation and analysis, often used for data preprocessing.

- scikit-learn: A comprehensive library for various machine learning algorithms, including classification, regression, clustering, and more.

- TensorFlow and PyTorch: Leading deep learning frameworks used for building and training neural networks.

- Keras: An easy-to-use high-level neural networks API built on top of TensorFlow and Theano (now integrated with TensorFlow as tf.keras).

2. **Ease of Use**: Python's simple and clean syntax makes it easy to read and write, reducing the learning curve for newcomers to machine learning. Its readability also facilitates better collaboration among team members.

3. **Community and Documentation:** Python has a vast and active community of data scientists, machine learning researchers, and developers. This community actively contributes to the development of machine learning tools, libraries, and frameworks. Additionally, there is extensive documentation available for most popular Python libraries, making it easier to get started and find solutions to common problems.

4. **Versatility and Integration:** Python is a versatile language that can be used for various tasks beyond machine learning. Its ability to integrate well with other languages and technologies makes it a preferred choice for incorporating machine learning models into larger applications.

5. **Data Processing and Visualization:** Python's ecosystem includes libraries like pandas and matplotlib, which enable efficient data preprocessing, cleaning, and visualization. These steps are essential for preparing the data before feeding it into machine learning models.

6. **Support for Deep Learning**: With the rise of deep learning, Python has become the goto language for building, training, and deploying deep neural networks. TensorFlow and PyTorch, both primarily Python-based, are among the most popular frameworks for deep learning.

7. **Open-Source and Free:** Python is an open-source language, which means it is freely available for anyone to use and modify. This accessibility has contributed to its widespread adoption in academia and industry.

Due to these advantages and its strong community support, Python has become the de facto language for machine learning tasks. It has democratized access to machine learning tools and made it easier for individuals and organizations to leverage the power of machine learning in their projects and research.

1.8 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It is a popular tool among data scientists, researchers, and developers for interactive computing and data analysis.

Here's a brief overview of Jupyter Notebook's key features and how to use it:

1. **Notebook Interface:** When you start Jupyter Notebook, it opens in your web browser and provides a file browser to navigate your system's directories. You can create new notebooks or open existing ones from this interface.

2. **Cell-based Structure**: Notebooks are organized into cells, which can be of two types:

- Code Cells: Used for writing and executing code (Python, R, Julia, etc.).

- Markdown Cells: Used for writing formatted text, including headings, lists, images, and more, using the Markdown syntax.

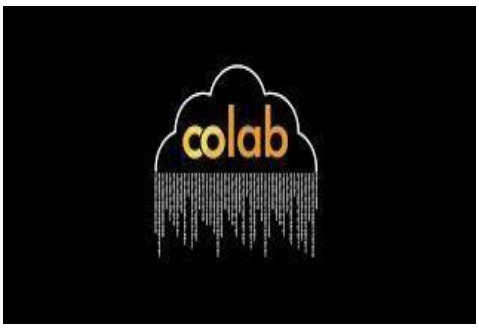
3. **Code Execution**: Code cells allow you to write and execute code in a step-by-step manner. To run a code cell, you can either click the "Run" button in the toolbar or use the keyboard shortcut "Shift + Enter." The output of the code execution is displayed directly below the cell.

4. **Kernel**: A notebook runs a kernel, which is essentially the computational engine that executes the code within the notebook. For example, if you write Python code, the notebook uses a Python kernel to execute it. You can switch kernels to work with different programming languages.

5. **Saving and Exporting**: Notebooks are automatically saved, but you can also manually save them by clicking the "Save" button or using the keyboard shortcut "Ctrl + S" (or "Cmd + S" on Mac). You can export notebooks to various formats, such as HTML, PDF, or plain Python scripts.

6. **Rich Media Support:** Notebooks allow you to embed images, videos, audio, and other media directly in the cells. This is particularly useful when creating interactive data visualizations.

1.9 Google Colab:



Google Colab (short for Collaboratory) is a cloud-based platform provided by Google that offers a Jupyter Notebooklike environment with access to free GPU and TPU (Tensor Processing Unit) resources. Colab is especially popular among data scientists, machine learning researchers, and students due to its ease of use, free access to computational resources, and the ability to collaborate and share notebooks easily.

Features of google colab

1. **Jupyter Notebook Environment**: Colab provides a Jupyter Notebook environment similar to what you would find in Jupyter Notebook. It supports both code cells (for writing and executing code) and markdown cells (for formatted text).

2. **Free GPU and TPU Support**: One of the major advantages of Colab is the provision of free access to GPUs and TPUs. This is particularly useful for training deep learning models that require substantial computational power.

3. **Cloud-based:** Colab runs entirely in the cloud, which means you don't need to install anything on your local machine. You can access your notebooks from any device with an internet connection and a web browser.

4. **Collaboration**: Colab allows multiple users to collaborate on the same notebook simultaneously. You can share the notebook with others and work on it together in real-time, making it a valuable tool for team projects and remote collaborations.

5. **Pre-installed Libraries**: Colab comes with many popular Python libraries pre-installed, such as TensorFlow, PyTorch, NumPy, pandas, matplotlib, and more. This saves time on setup and configuration, enabling you to focus on your analysis or machine learning tasks.

6. **File Storage:** Colab provides limited file storage in Google Drive. You can save and load files directly from your Google Drive, making it easy to keep your data and notebooks organized.

7. **Code Snippets and Examples:** Colab provides a range of code snippets and examples for different use cases, which can help users get started quickly and learn new techniques.

CHAPTER 2 – TRAINING WORK UNDERTAKEN

2.1 Project introduction – Crop Prediction (Agriculture production optimization)

Crop prediction , also known as agricultural yield forecasting ,is crucial aspect of modern agriculture that involves estimating the potential output of various crops based on specific soil

content requirements . It plays a vital role in agricultural planning , food security , and economic decision – making for farmers, governments ,and stakeholders within the agricultural sector.

The primary goal of crop prediction is to recommend the most suitable crops for a given region based on the soil's nutrient content and other relevant factors. By accurately predicting the best crops to plant, farmers can make informed decisions regarding planting, irrigation ,fertilization, and pest control ,which can optimize production and reduce potential losses.

Crop prediction is a complex task that relies on the integration of various data sources, including historical agricultural data, weather patterns, soil characteristics, and satellite imagery. Machine learning algorithms and statistical models are often employed to analyze these datasets and extract patterns and relationships that can be used to make predictions.

Benefits of accurate crop prediction include:

- Improved Resource Management: Farmers can optimize the use of resources such as water, fertilizers, and pesticides based on recommended crops, leading to cost savings and reduced environmental impact.
- Mitigating Food Shortages: Governments and organizations can use crop prediction to identify regions at risk of food shortages and take proactive measures to address potential food security challenges.
- Risk Management for Farmers: Knowing the best crops to plant in advance helps farmers manage financial risks and plan for market fluctuations.
- Sustainable Agriculture: By recommending crops that are well-suited to the soil's nutrient content, agricultural practices can be more sustainable, minimizing waste and environmental damage.

Overall, crop prediction based on soil content data is a powerful tool that empowers farmers and decision-makers with valuable insights to ensure the efficient and sustainable production of crops, contributing to food security and economic stability. This technique can be very well adapted to predict suitable crops, ensuring the efficient use of resources and enhancing agricultural productivity.

2.2 Libraries used

2.2.1 Pandas - Pandas is a popular Python library for data manipulation and analysis. It provides data structures for handling tabular and timeseries data, including the Data Frame and Series objects. Pandas offers a wide range of functions for data cleaning, filtering, grouping, merging, and reshaping. It also supports data visualization and integration with other Python libraries, such as NumPy and Matplotlib. Pandas' intuitive and powerful API has made it a go-to library for data analysts and scientists.

2.2.2 Numpy - NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object. Other mathematical operations can be performed using this but simply speaking we just need it to convert our images into some form of an array so that we can store the model that has been trained.

2.2.3 sklearn - scikit-learn (often abbreviated as sklearn) is a popular and widely-used machine learning library for Python. It provides a rich set of tools for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and preprocessing of data. scikit-learn is built on top of other foundational Python libraries like NumPy and SciPy, making it a fundamental part of the Python data science ecosystem.

2.3 Dataset

We provided some general information about milk quality prediction datasets and it typically contain.

This dataset is manually collected from observations. It helps us to build machine learning models to predict the quality of milk.

This dataset consists of 7 independent variables ie. pH, Temperature, Taste, Odor, Fat, Turbidity, and Color.

Generally, the Grade or Quality of the milk depends on these parameters. These parameters play a vital role in the predictive analysis of the milk.

Link of dataset used – https://www.kaggle.com/datasets/cpluzshrijayan/milkquality

Usage

The target variable is nothing but the Grade of the milk. It can be

Target

Low (Bad)

Medium (Moderate)

High (Good)

If Taste, Odor, Fat, and Turbidity are satisfied with optimal conditions then they will assign 1 otherwise 0.

Temperature and ph are given their actual values in the dataset.

We have to perform data preprocessing, and data augmentation techniques to build statistical and predictive models to predict the quality of the milk.

ALL INFORMATION ABOUT DATASET

```
mq.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1059 entries, 0 to 1058
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   pH           1059 non-null   float64
 1   Temprature   1059 non-null   int64  
 2   Taste        1059 non-null   int64  
 3   Odor         1059 non-null   int64  
 4   Fat          1059 non-null   int64  
 5   Turbidity    1059 non-null   int64  
 6   Colour       1059 non-null   int64  
 7   Grade        1059 non-null   object 
dtypes: float64(1), int64(6), object(1)
memory usage: 66.3+ KB
```

2.4 Training process

- First step is to upload required libraries in jupyter notebook that will be used for milk quality prediction.

```
import pandas as pd
import numpy as np
```

- Then we will describe our data and check the minimum and maximum values of different columns.

```
mq.describe()
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour
count	1059.000000	1059.000000	1059.000000	1059.000000	1059.000000	1059.000000	1059.000000
mean	6.630123	44.226629	0.546742	0.432483	0.671388	0.491029	251.840415
std	1.399679	10.098364	0.498046	0.495655	0.469930	0.500156	4.307424
min	3.000000	34.000000	0.000000	0.000000	0.000000	0.000000	240.000000
25%	6.500000	38.000000	0.000000	0.000000	0.000000	0.000000	250.000000
50%	6.700000	41.000000	1.000000	0.000000	1.000000	0.000000	255.000000
75%	6.800000	45.000000	1.000000	1.000000	1.000000	1.000000	255.000000
max	9.500000	90.000000	1.000000	1.000000	1.000000	1.000000	255.000000

- Then we will check the datatypes of the columns in the dataset

```
mq.dtypes
```

```
pH             float64
Temprature       int64
Taste            int64
Odor             int64
Fat              int64
Turbidity        int64
Colour           int64
Grade           object
dtype: object
```

  - Now we will check how many entries are there in Grade column

```
mq['Grade'].value_counts()
```

```
Grade
low       429
medium    374
high      256
Name: count, dtype: int64
```

- Now we will replace the strings into integer so that our whole dataset will be converted into numeric form.

```
# change the grades into [0,1,2]
mq['Grade'] = mq['Grade'].replace(['low','medium','high'],[0,1,2])
```

- Then we will divide all columns into input and output variables ‘x’ and ‘y’ respectively

```
x=mq.iloc[:,:-1]
y=mq.iloc[:,-1]
```

```
x.head()
```

	pH	Temprature	Taste	Odor	Fat	Turbidity	Colour
0	6.6	35	1	0	1	0	254
1	6.6	36	0	1	0	1	253
2	8.5	70	1	1	1	1	246
3	9.5	34	1	1	0	1	255
4	6.6	37	0	0	0	0	255

```
y.head()
```

```
0    2
1    2
2    0
3    0
4    1
Name: Grade, dtype: int64
```

- X is our input
- Y is our output

- Now we split our dataset for training and testing.

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

- Now we use the DecisionTreeClassifier algorithm

```
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)
```

```
▾ DecisionTreeClassifier
DecisionTreeClassifier()
```

- To check the accuracy,we will perform given function below

```
from sklearn.metrics import accuracy_score
y_pred = dt.predict(x_test)
accuracy_score(y_test,y_pred)
```

```
0.9905660377358491
```

- Prediction

```
# predict the test
y_pred = dt.predict([[6.6,35,1,0,1,0,254]])
y_pred
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
  warnings.warn(
array([2])
```

NOW WE ARE DONE WITH BASIC ALGORITHM

CHAPTER - 4

Conclusion

1. **Improved Accuracy**: Machine learning models have demonstrated improved accuracy in predicting milk quality compared to traditional methods, leveraging large datasets and sophisticated algorithms.
2. **Real-Time Monitoring**: These models enable real-time monitoring and prediction, allowing for timely interventions to ensure high milk quality
3. **Cost-Effectiveness**: Automation of milk quality prediction reduces the need for manual testing, leading to cost savings in the dairy industry.
4. **Scalability**: Machine learning systems can easily scale to handle large volumes of data from multiple sources, making them suitable for large dairy farms and cooperatives.
5. **Data-Driven Decisions**: The adoption of machine learning fosters a data-driven approach, helping stakeholders make informed decisions based on predictive analytics.
6. **Enhanced Food Safety**: By accurately predicting potential quality issues, machine learning contributes to enhanced food safety and consumer confidence in dairy product

Future Scope

1. **Integration with IoT**: Combining machine learning with IoT devices for continuous data collection can further enhance prediction accuracy and provide real-time insights.

2. **Advanced Algorithms**: Developing and employing more advanced machine learning algorithms, such as deep learning and reinforcement learning, could improve prediction capabilities and handle more complex patterns.

3. **Personalized Models**: Creating personalized prediction models for individual farms or regions to account for local variations in milk quality factors.\

4. **Blockchain for Traceability**: Integrating blockchain technology to ensure traceability and transparency in the milk supply chain, enhancing trust and verification processes.

5. **Predictive Maintenance**: Using predictive analytics for equipment maintenance in dairy farms to prevent malfunctions that could affect milk quality.
6. **Sustainability Metrics**: Incorporating sustainability metrics into the prediction models to monitor and improve the environmental impact of dairy farming practices.

7. **Cross-Industry Applications**: Applying these predictive models to other agricultural products, improving overall food quality and safety standards across the industry.

8. **Collaborative Platforms**: Developing collaborative platforms where farmers, scientists, and technologists can share data and insights to continuously improve predictive models.

9. **Regulatory Compliance**: Enhancing machine learning models to ensure they meet evolving regulatory standards and guidelines for milk quality.

10. **Consumer Applications**: Creating consumer-facing applications that provide transparency and information about milk quality, fostering greater consumer trust and engagement.

RESULT

- Here, shows real life model where we give some inputs and submit it and obtain result like ‘Low’,”Medium’,’High’

  Link for model – https://milk-quality-prediction.glitch.me