**GENE ANALYSIS FOR EARLY CANCER DETECTION**

### A Minor Project Report

*Submitted by*

**REEBA SEBASTIAN [ RA2011027010072]**

**ALTHAF KADER   [RA2011027010073]**

*Under the Guidance of*

**Dr. D. HEMAVATHI**

(Associate Professor, Department of Department of Data Science and Business Systems)

*In partial fulfillment of the requirements for the degree of*

### BACHELOR OF TECHNOLOGY

### COMPUTER SCIENCE ENGINEERING

with specialization in BIG DATA ANALYTICS

****

**DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS**

**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**KATTANKULATHUR – 603 203**

#### NOVEMBER 2023

### Annexure II

Department of Data Science and Business Systems

**SRM Institute of Science & Technology Own Work\* Declaration Form**

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

**Degree/Course : B.Tech in Computer Science and Engineering with a specialization in Big Data Analytics**

**Student Name : Althaf Kader, Reeba Sebastian**

**Registration Number : RA2011027010073, RA2011027010072**

**Title of Work : Gene analysis for early cancer detection**

We hereby certify that this assessment compiles with the University’s Rules and Regulations relating to Academic misconduct and plagiarism\*\*, as listed in the University Website, Regulations, and the Education Committee guidelines.

We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

* Clearly references / listed all sources as appropriate
* Referenced and put in inverted commas all quoted text (from books, web, etc)
* Given the sources of all pictures, data etc. that are not my own
* Not made any use of the report(s) or essay(s) of any other student(s) either past or present
* Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
* Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

|  |
| --- |
| **DECLARATION:** |
| I am aware of and understand the University’s policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above. |
| Reeba Sebastian Althaf Kader  RA2011027010072 RA2011027010073 |

BONAFIDE CERTIFICATE

Certified that this project report titled “**GENE ANALYSIS FOR EARLY CANCER DETECTION”** is the bonafide work of **“REEBA SEBASTIAN [RA2011027010072] and ALTHAF KADER [RA2011027010073]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr.D.Hemavathi Dr. M.Lakshmi

**GUIDE HEAD OF THE DEPARTMENT**

Associate Professor Dept. of DSBS

Dept. of DSBS

Signature of Internal Examiner Signature of External Examiner

**ABSTRACT**

This thesis presents an innovative approach to early cancer detection through gene analysis, employing a classification model that integrates Multiclass Support Vector Machine (SVM), Random Forest, and Deep Neural Networks. The study focuses on the critical problem of early cancer diagnosis, a pressing need in the medical field, due to the increasing prevalence and diversity of cancer types.

The main approach includes creating and contrasting three machine learning models, namely Multiclass SVM, Random Forest, and Deep Neural Networks. Each model is meticulously designed to classify input data into five distinct cancer types. This method enables a thorough assessment of the accuracy, specificity, and sensitivity of the models when it comes to detecting cancer.

The broader implications of this research for humanity are significant. Early and accurate cancer detection is crucial for effective treatment and can dramatically improve patient survival rates. By enabling earlier intervention, these models can reduce the burden of cancer on individuals and healthcare systems globally. Furthermore, this approach promises a shift towards more personalized medicine, where treatments can be tailored to individual genetic profiles, enhancing treatment efficacy and reducing side-effects.

In conclusion, this thesis highlights the efficacy of machine learning techniques in early cancer detection, promising significant improvements in patient outcomes. By facilitating early and accurate diagnoses, these models could revolutionize cancer care, leading to personalized treatments and reduced healthcare burdens, thereby making a profound contribution to the welfare of humanity.

**ACKNOWLEDGEMENT**

We express our humble gratitude to Dr C. Muthamizhchelvan, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, Dr T.V. Gopal, for his invaluable support.

We wish to thank Dr Revathi Venkataraman, Professor & Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, Dr M. Lakshmi Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our program coordinators Dr. G. Vadivu, Professor, Department of Data Science and Business Systems, SRM Institute of Science and Technology, for her inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, Dr. S Sharanya, Assistant Professor, Data Science and Business System, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to my guide, Dr. D Hemavathi, Associate Professor, Data Science and Business Systems SRM Institute of Science and Technology, for providing me with an opportunity to pursue my project under his/her/their mentorship. He/She/They provided me with the freedom and support to explore the research topics of my interest. Their passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank the Data Science and Business Systems staff and students, SRM Institute of Science and Technology, for their help during our project. Finally, we would like to than parents, family members, and friends for their unconditional love, constant support, and encouragement.

Althaf Kader [RA2011027010073]

Reeba Sebastian[RA2011027010072]

# TABLE OF CONTENTS

CHAPTER NO. TITLE PAGE NO.

ABSTRACT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| 1. | INTRODUCTION | 1 |
|  | 1.1 GENERAL | 1 |
|  | 1.2 PURPOSE | 2 |
|  | 1.3 SCOPE | 2 |
|  | 1.4 PROBLEM STATEMENT | 3 |
|  | 1.5 MOTIVATION | 4 |
| 2 | LITERATURE SURVEY | 5 |
| 3 | EXISTING SYSTEM | 7 |
|  | 3.1 ARCHITECTURE DIAGRAM | 9 |
| 4 | REQUIREMENTS | 11 |
|  | 4.1 SOFTWARE REQUIREMENTS | 11 |
|  | 4.2 HARDWARE REQUIREMENTS | 14 |
| 5 | METHODOLOGY | 18 |
|  | 5.1 UML DIAGRAM | 19 |
| 6 | RESULT AND DISCUSSION | 21 |
| 7 | CONCLUSION AND FUTURE ENHANCEMENT | 50 |
|  | REFERENCES | 52 |

**LIST OF TABLES**

6.1 Different types of cancers 21

**LIST OF FIGURES**

1 Architecture Diagram 9

2 UML Diagram 19

3 Distribution of cancer 25

4 PCA Plot 29

5 LDA Plot 30

6 t-SNE Plot 32

7 Dimensionality reduction using LDA 34

8 K-means clustering with PCA 37

9 K-means clustering 39

10 Confusion matrix 48

11 Accuracy Plot 48

**ABBREVIATIONS**

**AI** Artificial Intelligence

**CPS** Cyber-Physical System

**GUI**  Graphical User Interface

**SVM** Support Vector Machine

**GA** Genetic Algorithms

**PCA** Principal Component Analysis

**LDA** Linear Discriminant Analysis

**t-SNE** t-Distributed Stochastic Neighbor Embedding

**LIST OF SYMBOLS**

^ Conjunction

**CHAPTER 1**

**INTRODUCTION**

**1.1 General**

This comprehensive report presents the findings and methodologies employed by the Indian Council of Medical Research (ICMR) in their groundbreaking study on gene analysis for the early detection of cancer. The study is a testament to the advancements in genomic research and its pivotal role in understanding and combating various cancer types. Through meticulous analysis of genetic data from hundreds of patients, this report not only sheds light on the complex interplay of genes in cancer development but also opens new avenues for early diagnosis and personalized treatment strategies.

The importance of this research lies in its potential to revolutionize the field of oncology. By decoding the genetic blueprint of cancers, such as breast, renal, colon, lung, and prostate cancer, the study aims to identify key genetic markers that could be pivotal in early detection. This approach is particularly significant given the multifactorial nature of cancer and the challenges posed by its early detection and treatment.

The report navigates through the intricate process of data collection, analysis, and interpretation. It encompasses a vast array of genetic data, robust analytical techniques, and innovative research methodologies. The study's implications extend beyond the realm of cancer research, offering insights into genetic engineering, molecular biology, and clinical diagnostics. It represents a confluence of interdisciplinary research efforts, showcasing the potential of genetic analysis in transforming healthcare and medicine.

**1.2 Purpose**

The primary purpose of this extensive study is to delve into the genetic basis of various cancers that have increasingly become a cause for concern in recent years. By analyzing genetic data from a significant number of individuals diagnosed with cancer, the study aims to unravel the genetic intricacies that lead to the development of these malignancies. This endeavor is crucial in identifying potential genetic markers that could be used for early detection of cancers, a critical factor in improving treatment outcomes and survival rates.

The study is driven by the urgent need to address the growing incidence of cancers worldwide. It seeks to contribute significantly to the field of precision medicine, where treatments can be tailored based on an individual's genetic makeup. By pinpointing specific genes associated with each type of cancer, the study opens up possibilities for early intervention strategies, potentially shifting the focus from treatment to prevention.

Furthermore, the purpose extends to enhancing the understanding of oncologists, geneticists, and researchers about the genetic underpinnings of cancer. This knowledge is crucial for developing targeted therapies, improving diagnostic techniques, and fostering a deeper understanding of how genetic mutations influence cancer progression. The study's findings could lead to a paradigm shift in how cancer is perceived and treated, heralding a new era in cancer care that is more effective, personalized, and preventive.

**1.3 Scope**

The scope of this research encompasses a comprehensive analysis of 802 samples from individuals diagnosed with five different types of cancer: Breast Cancer (BRCA), Renal Cancer (KIRC), Colon Cancer (COAD), Lung Cancer (LUAD), and Prostate Cancer (PRAD). Each sample is a rich repository of information, containing expression values of over 20,000 genes, offering a vast dataset for thorough genetic analysis.

This study is not just about analyzing a large volume of data; it is about dissecting and understanding the complexity of cancer at the genetic level. The scope includes employing advanced bioinformatics tools and statistical methods to decode the massive amounts of genetic information. The study aims to filter out the noise and identify the crucial genetic signals that could be indicative of cancer development.

Moreover, the scope involves the application of various sophisticated data analysis techniques. These include dimensionality reduction to manage the complexity of the data, clustering methods to discern patterns and similarities in gene expression, and building robust classification models to differentiate between various cancer types. This multifaceted approach ensures a comprehensive understanding of the genetic factors involved in cancer, paving the way for groundbreaking findings in early cancer detection.

**1.4 Problem Statement**

The increasing prevalence of various cancer types, such as breast, renal, colon, lung, and prostate cancer, poses a significant challenge to the medical community. Despite advancements in medical science, the detection and treatment of cancer at an early stage remain a critical issue. The complexity of cancer, coupled with its often asymptomatic nature in the early stages, makes early detection challenging yet crucial for successful treatment outcomes.

The problem this study addresses is two-fold: firstly, to identify the specific genes responsible for different cancer types, and secondly, to leverage this information for early detection. The identification of these genes is not just a matter of scientific inquiry but a crucial step towards understanding the biological mechanisms that lead to cancer. This understanding is essential for developing targeted interventions that can prevent or halt the progression of cancer at its onset.

The challenge lies in analyzing and interpreting the vast and complex genetic data associated with cancer. With each sample containing expression values of over 20,000 genes, the task of pinpointing the relevant genetic markers is akin to finding a needle in a haystack. This problem is compounded by the variability and uniqueness of genetic expressions in different individuals, making the task of generalizing findings across populations a daunting one.

**1.5 Motivation**

The motivation behind this extensive research is rooted in the pressing need to combat the growing burden of cancer globally. With millions of people affected by cancer each year, and the number only rising, there is an urgent need for innovative approaches to cancer detection and treatment. This study is driven by the hope that understanding the genetic basis of cancer can lead to more effective, timely, and personalized medical interventions.

At the heart of this motivation is the belief that early detection is key to improving cancer survival rates. By identifying genetic markers that indicate a predisposition to cancer or its early onset, there is potential to save countless lives through early intervention. This research is not just about advancing scientific knowledge; it is about making a tangible difference in the lives of those affected by cancer.

The study is also motivated by the desire to contribute to the field of precision medicine, where treatments are tailored to the individual based on their genetic makeup. This approach has the potential to revolutionize cancer treatment, making it more effective and reducing the likelihood of adverse side effects. By identifying the specific genetic mutations associated with different cancers, this research paves the way for the development of targeted therapies that can precisely target cancerous cells without harming healthy ones.

**CHAPTER 2**

**LITERATURE STUDY**

The field of gene analysis for early cancer detection has seen significant advancements, as evidenced by a range of studies and developments highlighted in recent literature.

In the realm of computational methods, the application of genetic algorithms in cancer detection and prediction has proven to be a promising approach [1]. These algorithms, known for their efficiency in handling complex problems, have been utilized to discern patterns in genetic data that might be indicative of cancerous changes. This aligns well with the objectives of gene analysis in early cancer detection, where identifying subtle genetic variations is crucial.

The evolution of DNA typing has also been a cornerstone in understanding genetic predispositions to cancer [2]. The historical perspective provided by this advancement underscores the importance of genetic analysis in both human and non-human samples, highlighting how techniques have evolved to become more precise and informative in identifying genetic markers associated with cancer.

The general review of genetic algorithms [3] further contextualizes their role in the broader spectrum of computational biology. The adaptability and robustness of these algorithms make them particularly suitable for the complex task of analyzing high-dimensional genetic data, a key component in early cancer detection.

Human molecular genetics and genomics have seen important advances, particularly with the advent of CRISPR-Cas9 and other gene-editing technologies [4]. These developments have opened up new possibilities in understanding the genetic basis of cancer, allowing for more targeted approaches in both detection and treatment.

The integration of artificial intelligence (AI) in early cancer diagnosis has been another significant leap forward [5]. AI's capability to analyze large datasets efficiently complements genetic analysis by providing the tools necessary to interpret complex genetic information, thereby enhancing the accuracy and speed of early cancer detection.

Finally, the historical and future perspectives on early cancer detection [6] provide a comprehensive view of the field. It emphasizes the critical role of emerging technologies and methodologies in improving the detection and subsequent treatment of cancer. This highlights the continuous need for innovation in gene analysis techniques to keep pace with the evolving nature of cancer biology.

In summary, the literature underscores a multidisciplinary approach in the fight against cancer, combining advancements in computational algorithms, genetic analysis, AI, and molecular genetics. This confluence of technologies and methodologies is pivotal in enhancing the effectiveness of early cancer detection, directly aligning with the goals of the project on gene analysis for early cancer detection.

**CHAPTER 3**

**EXISTING SYSTEM**

Conventional cancer screening programs have been pivotal in the early detection of malignancies, yet they face significant limitations that affect their overall efficiency. One of the primary challenges is the limited range of cancers for which screening is recommended. This narrow focus leaves a substantial number of malignancies undetected in their early stages. Additionally, these programs often rely on multimodal testing methods, which unfortunately contribute to a high cumulative false-positive rate. False positives not only cause unnecessary anxiety and stress for patients but also lead to an increase in the iatrogenic burden, where individuals undergo further invasive procedures that may not have been needed. This scenario underscores the need for more efficient and comprehensive screening strategies that can encompass a broader spectrum of cancers while minimizing the rate of false positives.

The Emergence of Multi-Cancer Early Detection Technologies: In response to these challenges, the concept of a single test for multi-cancer early detection (stMCED) has gained traction as a promising alternative to traditional screening methods. This approach aims to enhance the efficiency of cancer detection by screening for multiple types of cancer simultaneously in asymptomatic individuals. The appeal of stMCED lies in its potential to capitalize on the aggregated prevalence of different cancers, coupled with a fixed specificity, to improve the positive predictive value of the screening process. This value is crucial in determining the accuracy of a test and balancing the potential harms and benefits. The development of stMCED technologies has seen a recent surge, driven by the promise of improving clinical outcomes and the overall cost-benefit of cancer screening programs. However, these technologies face unique challenges in their development, particularly in terms of effectively integrating them into existing clinical workflows and screening guidelines.

Advancements in Circulating Tumor DNA Analysis and Epigenetic Biomarkers: One of the most promising avenues in the development of stMCED methodologies is the analysis of circulating tumor DNA (ctDNA). This approach involves detecting DNA methylation biomarker fingerprints, which are indicative of malignancy etiology and progression. The potential of ctDNA lies in its ability to provide tissue- and cancer-type specific information, making it a powerful tool for early-stage cancer detection. Furthermore, utilizing panels of epigenetic biomarkers through blood testing could potentially allow for the identification of the tumor's origin, which is invaluable for subsequent clinical decision-making and patient care. This method represents a significant advancement over traditional screening methods, offering a minimally invasive, more comprehensive, and potentially more accurate approach to cancer detection. As such, the integration of ctDNA analysis into stMCED methodologies is a critical area of focus, promising to revolutionize the landscape of cancer screening and patient management.

**3.1 ARCHITECTURE DIAGRAM**

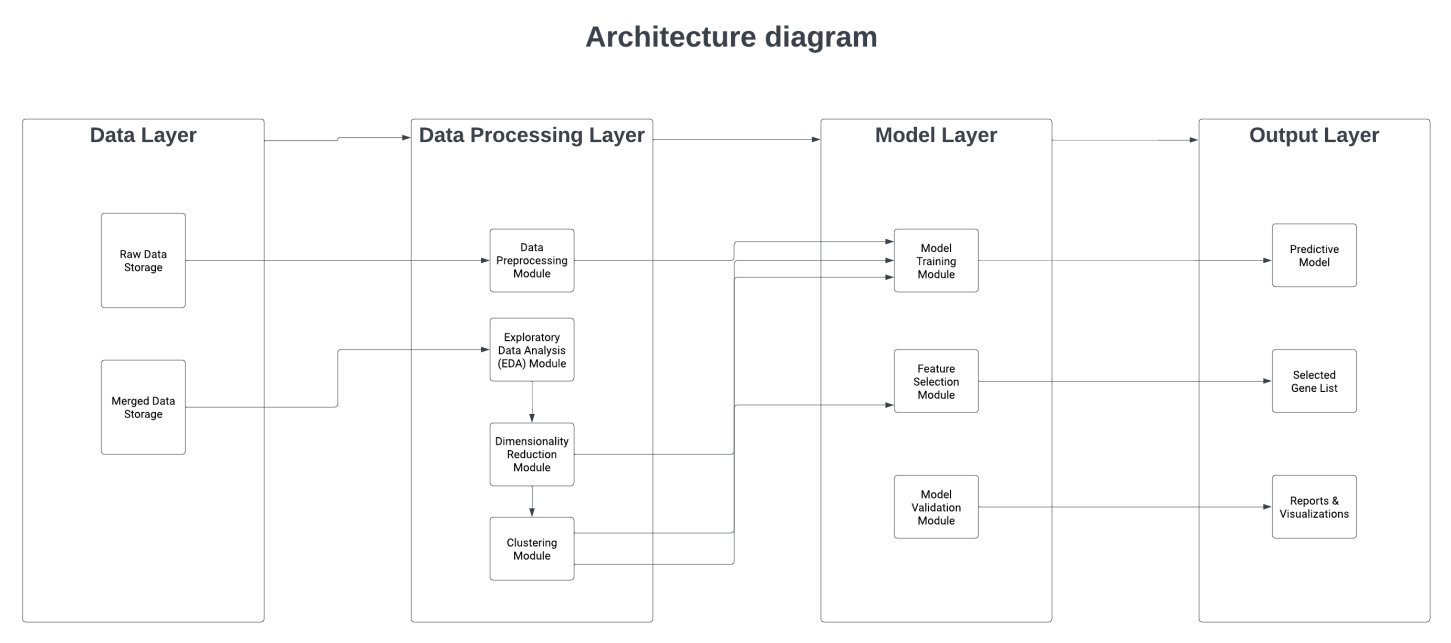


FIG 3.1: ARCHITECTURE DIAGRAM

**FIG 3.1:**Architecture diagram for Gene Analysis for Early Cancer Detection

**Design of modules**

The process begins with the input of a dataset .The dataset encompasses 802 samples, each representing unique genetic expressions from individuals diagnosed with one of five cancer types: BRCA, KIRC, COAD, LUAD, and PRAD. Each sample contains over 20,000 gene expression values, presenting a rich yet complex dataset for analysis. The initial step involves exploratory data analysis, including merging datasets and creating hierarchically-clustered heatmaps, along with performing null-hypothesis testing to understand the underlying patterns in the data. A crucial aspect of the model design is dimensionality reduction, where techniques like PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), and t-SNE (t-Distributed Stochastic Neighbor Embedding) are employed to distill the dataset down to a manageable number of relevant genes for each cancer type.

**1) Dataset Preparation and Exploratory Data Analysis**:

Analyze 802 samples, each corresponding to an individual diagnosed with one of the cancer types: BRCA, KIRC, COAD, LUAD, or PRAD.

**2) Dimensionality Reduction:**

Implement PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), and t-SNE (t-Distributed Stochastic Neighbor Embedding) to reduce the dataset to a more manageable size.

**3) Clustering Genes and Samples:**

Apply clustering techniques such as k-means, hierarchical, and mean shift clustering to the refined dataset.

**4)Building Classification Models:**

Develop classification models using multiclass SVM (Support Vector Machines), Random Forest, and Deep Neural Networks.

**5)Feature Selection:**

Apply feature selection algorithms, including forward selection and backward elimination.

**6)Validation of Selected Genes:**

Utilize statistical significance testing methods, such as t-tests for one vs. all and F-tests. Confirm the reliability and relevance of the identified genes in accurately classifying each cancer type.

**CHAPTER 4**

**REQUIREMENTS**

**SOFTWARE REQUIREMENTS:**

To execute and manage the project which involves the analysis of gene expression data for different types of cancer, a combination of specialized software and adequate hardware is essential.

1. Programming Languages and Environments:

Python: It is a versatile language widely used in data science and machine learning.

Key libraries include:

NumPy and Pandas for data manipulation.

Scikit -learn for accessible machine learning algorithms.

Tensor Flow or Keras for more complex, deep learning models.

Matplotlib and Seaborn for creating insightful data visualizations.

Python's extensive libraries and community support make it a top choice for data-driven projects.

R: Highly regarded for statistical analysis and graphical capabilities.

Particularly strong in specific bioinformatics applications and statistical modeling.

Offers robust package ecosystems like CRAN and Bioconductor, which provide tools tailored for a wide range of data analysis needs.

1. Data Analysis and Machine Learning Libraries:

Scikit - learn: A foundational library in Python for machine learning.

Offers a wide range of algorithms for classification, regression, clustering, and dimensionality reduction.

Known for its ease of use and ability to integrate seamlessly with other Python libraries.

TensorFlow/Keras: TensorFlow, with its high-level API Keras, is ideal for building and training deep learning models.

It supports complex neural network architectures and is flexible enough for research and production.

Biopython: A specialized library for computational biology and bioinformatics.

Provides tools for DNA and protein sequence analysis, alignment, and database access.

Matplotlib, Seaborn, Plotly: These libraries are vital for visualizing data and model outcomes.

Matplotlib and Seaborn are great for static plots, while Plotly offers interactive plotting capabilities.

1. Bioinformatics Tools (Optional):

BLAST, Bioconductor, etc.: These tools are essential for specific tasks in genomic analysis.

BLAST for sequence searching, Bioconductor for high-throughput genomic data analysis, among others.

1. Integrated Development Environment (IDE):

Jupyter Notebook or JupyterLab:

Ideal for exploratory data analysis and interactive computing.

Supports mixing code, visualizations, and documentation in a single document.

RStudio:

A comprehensive IDE for R, providing tools for plotting, debugging, and workspace management.

1. Version Control:

Git:

Essential for tracking changes in code, particularly in collaborative projects.

Facilitates branching, merging, and versioning of code repositories.

GitHub or GitLab:

Platforms for hosting code repositories.

Provide additional features like issue tracking, documentation, and continuous integration.

1. Database Management Systems (if handling large datasets):

SQL-based systems (MySQL, PostgreSQL, SQLite):

Ideal for structured data that fits well into tabular formats.

Supports complex queries and is essential for large-scale data management.

NoSQL databases (MongoDB):

More flexible in handling unstructured or semi-structured data.

Useful when dealing with diverse data types or rapidly changing schemas.

## 

## **HARDWARE REQUIREMENTS:**

1. Processor:

Multi-core CPU (Intel i7, i9, or equivalent):

Crucial for efficiently handling computationally intensive tasks like data processing, model training, and analysis.

A higher number of cores and threads facilitates parallel processing, which can significantly speed up computations.

Advanced CPUs also have better cache memory management, enhancing performance for data-intensive operations.

1. RAM:

Minimum 16GB, preferably 32GB or more:

Adequate RAM is essential for smooth multitasking and handling large datasets, which can be memory-intensive.

More RAM allows for larger portions of data to be loaded into memory at once, reducing reliance on slower disk reads.

Essential for running complex models, especially those in machine learning and deep learning that require substantial memory.

1. Storage:

SSD (Solid State Drive):

SSDs offer faster read/write speeds compared to traditional HDDs, accelerating data access and processing.

This speed is beneficial for tasks that involve frequent disk operations, like loading large datasets or handling database transactions.

Capacity needs to be considered based on the size of datasets; for extensive projects, a 1TB or larger SSD is recommended.

External or Cloud-based Storage:

External storage solutions are important for data backup, ensuring data safety in case of hardware failures.

Cloud-based storage offers scalability and remote accessibility, facilitating collaboration and providing an off-site backup option.

1. GPU:

High-performance GPU (like NVIDIA RTX or Tesla series):

GPUs are optimized for parallel processing, making them significantly faster than CPUs for specific tasks like training deep learning models.

They can handle thousands of threads simultaneously, accelerating computations in algorithms that support parallelization.

Essential for projects involving neural networks and large-scale simulations, where GPU acceleration can drastically reduce training and inference times.

1. Network Capabilities:

Fast Internet Connection:

A robust internet connection is crucial for accessing cloud computing resources, large databases, and online collaboration tools.

Enables smooth data transfer to and from cloud services, which is essential for cloud-based computing and storage.

Important for staying connected with team members, accessing online resources, and staying up-to-date with the latest software and updates.

1. Operating System:

Windows, Linux, or macOS:

Choice of OS depends on personal preference, software requirements, and compatibility with various tools and platforms.

Linux is often preferred for its stability, security, and better handling of server-side applications.

Windows and macOS offer user-friendly interfaces and broad compatibility with most mainstream applications and tools.

1. Cloud Computing Resources (Optional):

Platforms like AWS, Google Cloud, or Azure:

Provide scalable and flexible computing resources, suitable for handling very large datasets and computationally intensive models.

Offer a range of services like virtual machines, managed databases, and AI and machine learning tools.

Beneficial for projects requiring high computational power, which might be impractical or too expensive to set up on-premises.

**CHAPTER 5**

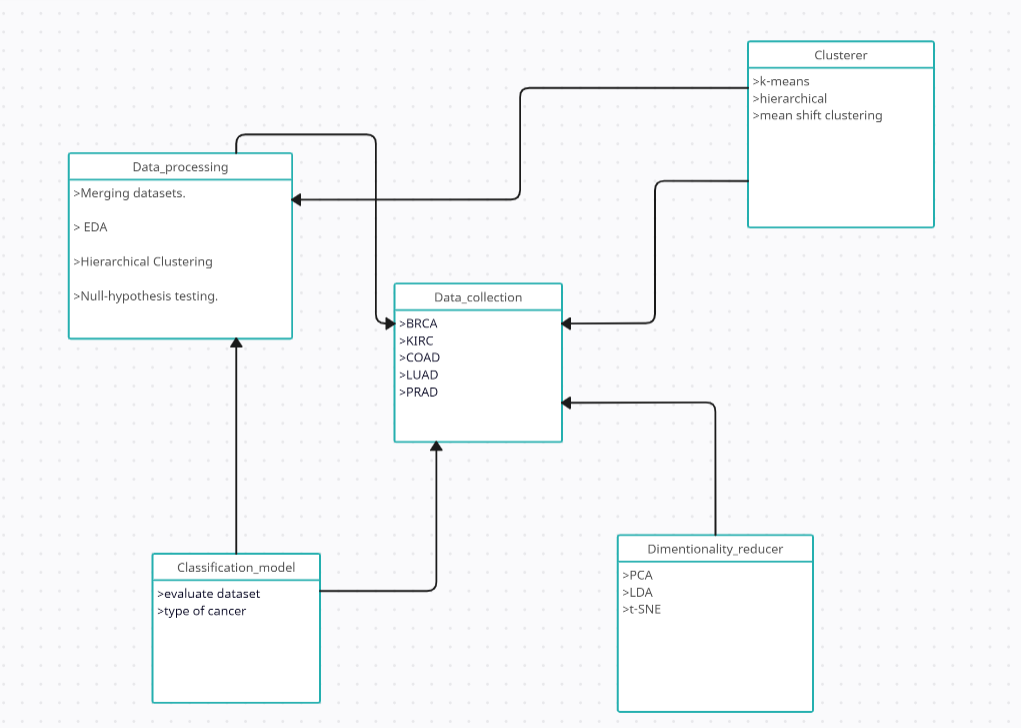
**METHODOLOGY**

The methodology for analyzing various types of cancers for gene analysis begins with a comprehensive dataset comprising 802 samples, each representing a unique case of cancer such as BRCA, KIRC, COAD, LUAD, and PRAD. Each sample within this dataset contains the expression values of more than 20,000 genes, providing a detailed genetic landscape of the cancer types under investigation. The initial phase of the analysis, exploratory data analysis, involves merging datasets and visualizing the combined data through hierarchically-clustered heatmaps, along with performing null-hypothesis testing to identify significant patterns.

Subsequently, the process focuses on dimensionality reduction using techniques like PCA, LDA, and t-SNE to distill the extensive gene expression data to a more manageable and relevant subset. This step aims to isolate a smaller set of attributes critical for the analysis of each specific cancer type. Following this, the methodology employs clustering techniques such as k-means, hierarchical, and mean shift clustering to categorize genes based on similar expression values across samples, as well as to group samples by cancer type.

The final and most crucial phase involves building robust classification models using multiclass SVM, Random Forest, and Deep Neural Network. These models are designed to classify the input data into the five identified cancer types accurately. Integral to this phase is the feature selection process, where techniques like forward selection and backward elimination are applied to refine the attributes identified during dimensionality reduction. The selected genes are then validated through statistical significance testing, including t-tests and F-tests, ensuring the robustness and accuracy of the model in identifying the genetic causes of each cancer type. This comprehensive methodology aims to lead to early identification of various cancers, potentially reducing the fatality rate associated with these diseases.

**UML DIAGRAM:**

****

**FIG 5.1:** UML DIAGRAM for Gene Analysis for Early Cancer Detection

The UML Activity Diagram for gene analysis is a structured representation of the sequence of activities involved in this complex process. At the onset, the Start Node marks the beginning of the workflow. The first major activity is Data Collection, where 802 samples corresponding to different cancer types (BRCA, KIRC, COAD, LUAD, PRAD) are gathered. Each sample contains extensive gene expression data, setting the foundation for the analysis.

Data Collection: Involves gathering 802 samples across various cancer types (BRCA, KIRC, COAD, LUAD, PRAD), each containing extensive gene expression data, essential for the analysis.

Data Preprocessing: Consists of merging datasets, performing Exploratory Data Analysis (EDA), creating hierarchically-clustered heatmaps for visualization, and conducting null-hypothesis testing to establish data patterns.

Decision Node for Dimensionality Reduction: A critical juncture where techniques like PCA, LDA, and t-SNE are employed to simplify the complex gene expression data while retaining significant information.

Clustering of Genes and Samples: Utilizes clustering methods (k-means, hierarchical, mean shift) to identify gene expression similarities and categorize samples according to cancer type.

Validation of Genes: Ensures the reliability and relevance of the model through statistical significance testing methods, including t-tests and F-tests.

**CHAPTER 6**

**RESULT AND DISCUSSION**

The dataset consists of 20563 columns and 802 rows consisting of the diseases mentioned below.

Below is an explanation of each cancer along with how these will be preprocessed to allow for better analysis and model building

**Table 6.1:**Different types of cancers

|  |  |
| --- | --- |
| Name of the cancer | Description |
| Renal cancer | Renal cancer, commonly known as kidney cancer, is a type of cancer that originates in the kidneys, the two bean-shaped organs responsible for filtering waste from the blood and producing urine. |
| Breast cancer | Breast cancer is a malignant tumor that originates in the cells of the breasts, commonly presenting as a lump or changes in breast shape or texture. |
| Colon cancer | Colon cancer, also known as colorectal cancer, is a malignancy that develops in the tissues of the colon (large intestine) and can spread to other parts of the body if left untreated. |
| Lung cancer | Lung cancer is a malignant tumor that originates in the lungs, often linked to smoking and exposure to carcinogens, and it can severely impair lung function and spread to other organs if not detected and treated early. |
| Prostate cancer | Prostate cancer is a common form of cancer in men, originating in the prostate gland, which is responsible for producing seminal fluid. |

1. **Exploratory Data Analysis**

This section contains the complete source code utilized for preprocessing the dataset, performing exploratory data analysis, and creating visualizations. The scripts are integral for understanding the initial stages of data handling, including merging different datasets and addressing issues like missing values or outliers.

CODE:

#Exploratory Data Analysis

importnumpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

importos

fordirname, \_, filenames in os.walk(r'C:\Users\Rozario\Downloads\icmrdata'): #path where dataset is stored on device

for filename in filenames:

print(os.path.join(dirname, filename))

import pandas as pd

importnumpy as np

importscipy.stats as stats

importmatplotlib.pyplot as plt

importseaborn as sns

fromsklearn.decomposition import PCA

fromsklearn.discriminant\_analysis import LinearDiscriminantAnalysis

colors = ['royalblue','red','deeppink', 'maroon', 'mediumorchid', 'tan', 'forestgreen', 'olive', 'goldenrod', 'lightcyan', 'navy']

vectorizer = np.vectorize(lambda x: colors[x % len(colors)])

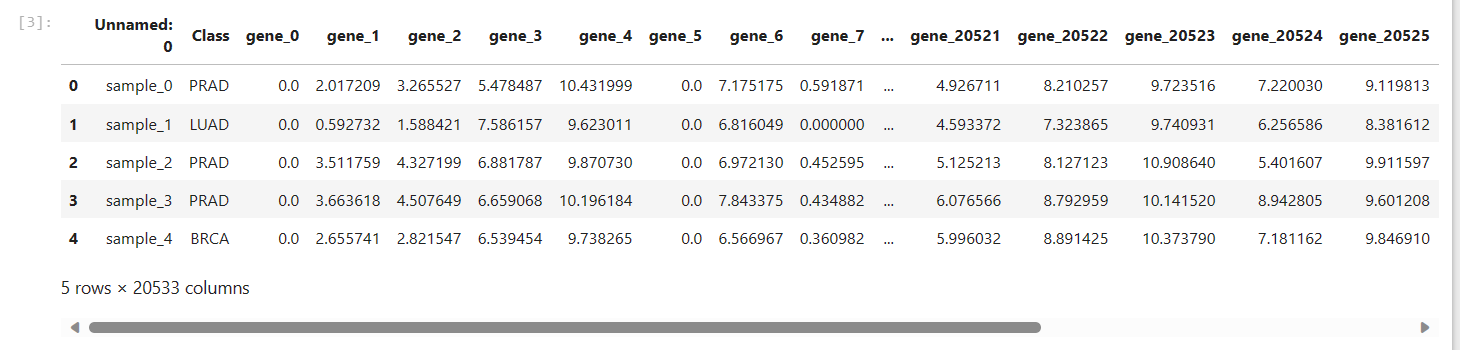
#merge the datasets

data = pd.read\_csv(r'C:\Users\Rozario\Downloads\icmrdata\data.csv')

label = pd.read\_csv(r'C:\Users\Rozario\Downloads\icmrdata\labels.csv')

df = pd.merge(label,data)

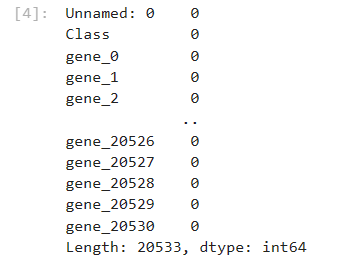
df.head()



Above output is the whole merged dataset with around 20K gene of 802 individuals with one of the five types of cancer.

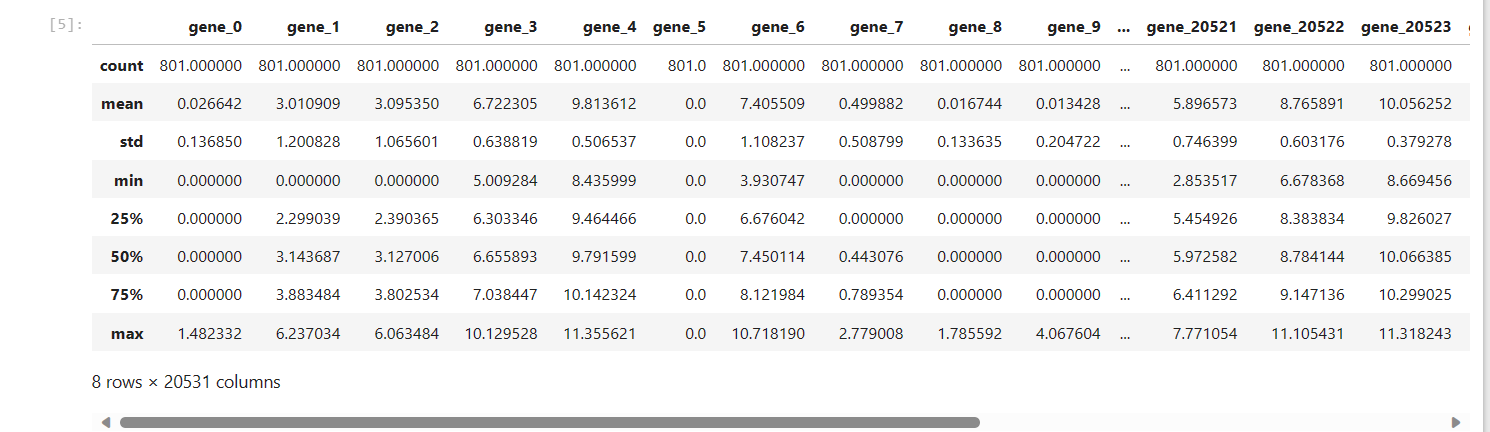
# Identifying and handling missing data. removes null values

df.isnull().sum()



Above figure shows the null entities in the dataset that are removed. This is a step in data cleaning.

df.describe()



print(df.dtypes)

#merge dataset into hirerarchially clustered heatmap

heatmap\_data = pd.pivot\_table(df, index=['Class'])

heatmap\_data.head()

sns.clustermap(heatmap\_data)

plt.savefig('heatmap\_with\_Seaborn\_clustermap\_python.jpg',

dpi=150, figsize=(8,12))

#paste heatmap1

sns.clustermap(heatmap\_data, figsize=(18,12))

plt.savefig('clustered\_heatmap\_with\_dendrograms\_Seaborn\_clustermap\_python.jpg',dpi=150)

#paste heatmap2

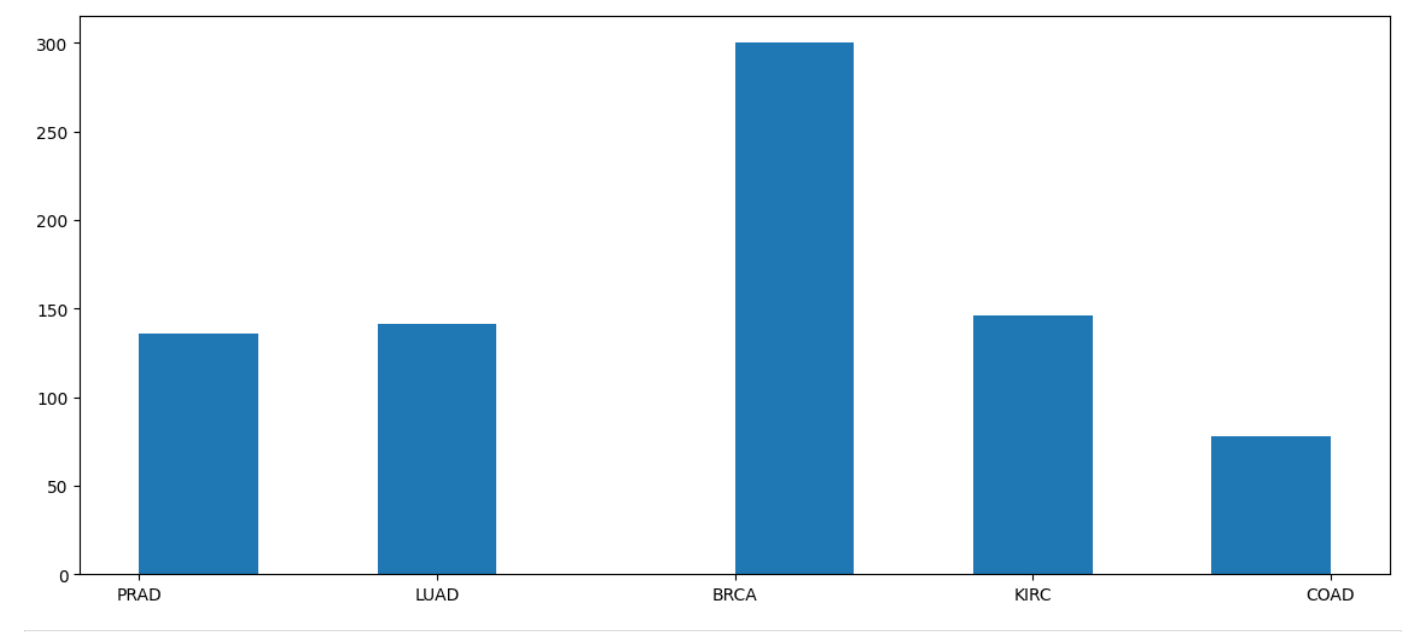
#Perform Null Hypothesis testing

#histogram to check if the data is normally distributed

plt.figure(figsize=(14,6))

plt.hist(df['Class'])

plt.show()



**FIG 6.1:**Distribution of cancer

Above figure shows the distribution of the five different types of cancers (Renal, Breast, Colon, Lung, Prostate) respectively over the sample 802 people with cancer.

non\_cat\_data = df.drop(['Unnamed: 0'], axis=1)

non\_cat\_data

#F-test

df\_f\_test=df

deff\_test(df\_f\_test,gene):

df\_anova = df\_f\_test[[gene,'Class']]

grps = pd.unique(df\_anova.Class.values)

grps

d\_data = {grp:df\_anova[gene][df\_anova.Class == grp] for grp in grps}

F, p = stats.f\_oneway(d\_data['LUAD'], d\_data['PRAD'], d\_data['BRCA'], d\_data['KIRC'], d\_data['COAD'])

print("p\_values:-",p)

if p<0.05:

print("reject null hypothesis")

else:

print("accept null hypothesis")

return

f\_test(df\_f\_test,"gene\_6")

f\_test(df\_f\_test,"gene\_20522")

f\_test(df\_f\_test,"gene\_5")

df\_cat\_data = df

df\_cat\_data['Class'] = df\_cat\_data['Class'].map({'PRAD': 1, 'LUAD': 2, 'BRCA': 3, 'KIRC': 4, 'COAD': 5})

df\_cat\_data = df\_cat\_data.drop(['Unnamed: 0'],axis=1)

1. **Dimensionality Reduction Techniques**

This part of the appendix shares scripts related to the implementation of dimensionality reduction techniques, namely Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and t-Distributed Stochastic Neighbor Embedding (t-SNE). Each sample has expression values for around 20K genes. However, it may not be necessary to include all 20K genes expression values to analyze each cancer type. Therefore, we will identify a smaller set of attributes which will then be used to fit multiclass classification models. So, the first task targets the dimensionality reduction using various techniques such as, PCA, LDA, and t-SNE It also includes any custom functions or algorithms developed for these analyses.

CODE:

#Dimensionality Reduction

#Dimensionality Reduction using PCA

# Define data

df\_pca = df.drop(['Unnamed: 0'], axis=1)

df\_pca = df\_pca.drop(['Class'], axis=1)

df\_pca.head()

df\_pca.values.shape

x\_pca = df\_pca.values

#Input: Complete dataset including all genes (20531)

#Scaling the data using standard scaler method

fromsklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_Scaled = scaler.fit\_transform(x\_pca)

X\_Scaled

#Perform PCA with n\_components=2

# Import PCA from sklearn and define the n\_components as 2

fromsklearn.decomposition import PCA

pca\_with\_2=PCA(n\_components=2)

#Perform fit transform on the scaled data

X\_pca\_with\_2 = pca\_with\_2.fit\_transform(X\_Scaled)

X\_pca\_with\_2.shape

X\_pca\_with\_2

# Put the data back on the 2 columns defined

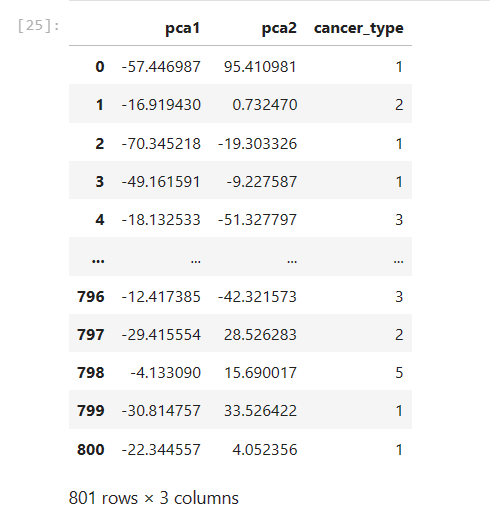
df\_pca = pd.DataFrame(X\_pca\_with\_2)

df\_pca.columns = ['pca1','pca2']

# Add the convereted categorical data for

df\_pca['cancer\_type']=df\_cat\_data['Class']

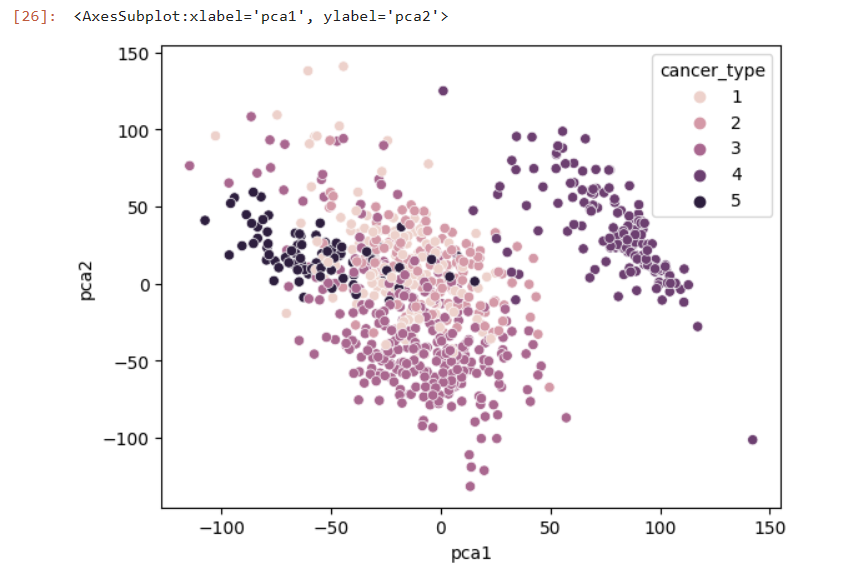
df\_pca



Above output shows the two PCA components pca1 and pca2 and the corresponding cancer type depicted from numerals 1 to 5.

# Present the data on the 5 clusters using seaborn maps

sns.scatterplot(x='pca1',y='pca2', hue = 'cancer\_type',data=df\_pca)



**FIG 6.2:** PCA plot

Above figure shows PCA dimensionality reduction output as a plot with components=2. This is done so as to  identify a smaller set of attributes because it is not be necessary to include all 20K gene expression values to analyze each cancer type.

#PCA with n\_components=.995

pca\_with\_995=PCA(.995)

X\_pca\_with\_995 = pca\_with\_995.fit\_transform(x\_pca)

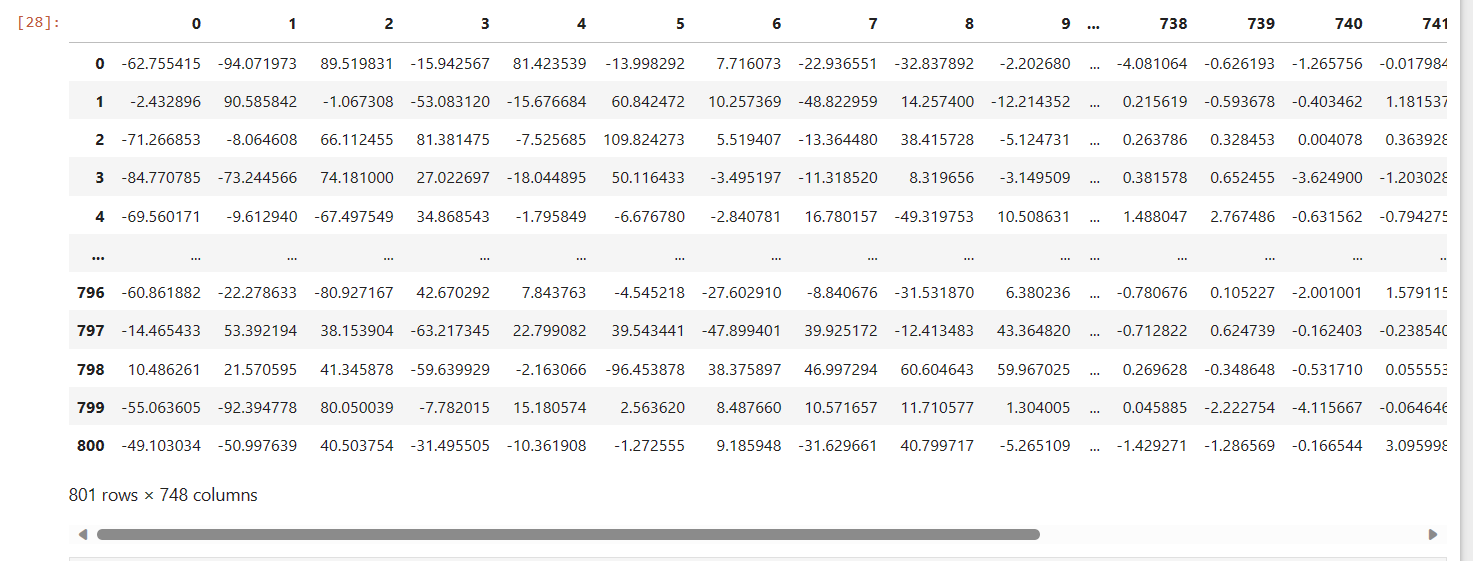
X\_pca\_with\_995.shape

X\_pca\_with\_995

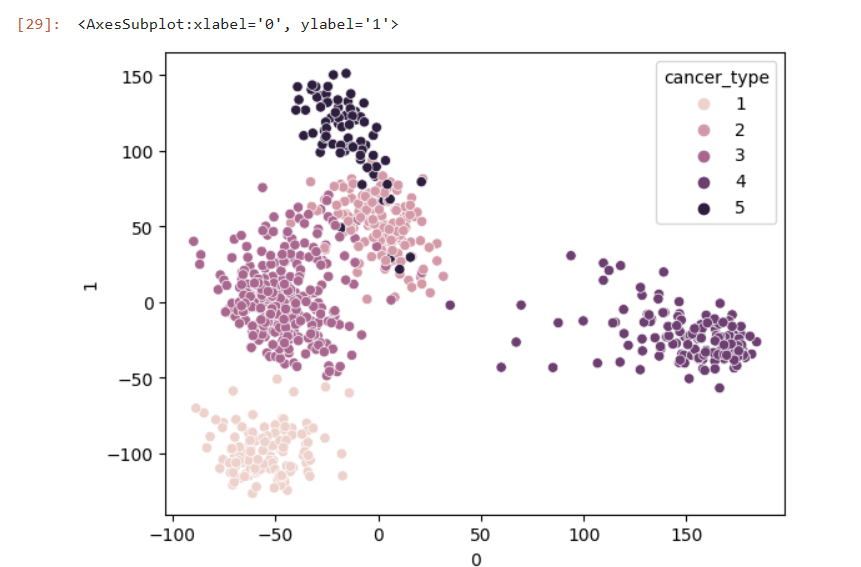
df\_pca\_995 = pd.DataFrame(X\_pca\_with\_995)

df\_pca\_995['cancer\_type']=df\_cat\_data['Class']

df\_pca\_995



sns.scatterplot(x=0,y=1,hue = 'cancer\_type', data=df\_pca\_995)



**FIG 6.3:**LDA Plot

Above figure shows LDA dimensionality reduction output as a plot. This is done so as to  identify a smaller set of attributes because it is not be necessary to include all 20K gene expression values to analyze each cancer type.

#Output: Selected Genes from each dimensionality reduction method

#Dimensionality reduction using t-SNE

df\_tsne\_data = df

non\_numeric = ['Unnamed: 0','Class']

df\_tsne\_data = df\_tsne\_data.drop(non\_numeric, axis=1)

df\_tsne\_data

# import t-SNE from sklearn

fromsklearn.manifold import TSNE

m = TSNE(learning\_rate=50)

tnse\_features = m.fit\_transform(df\_tsne\_data)

tnse\_features[1:4,:]

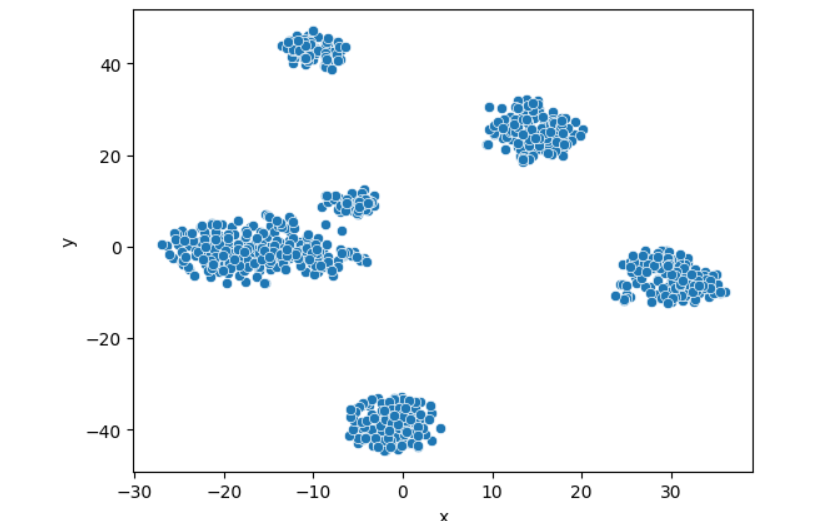
df\_tsne\_data['x'] = tnse\_features[:,0]

df\_tsne\_data['y'] = tnse\_features[:,1]

importseaborn as sns

sns.scatterplot(x='x',y='y',data=df\_tsne\_data)

plt.show()



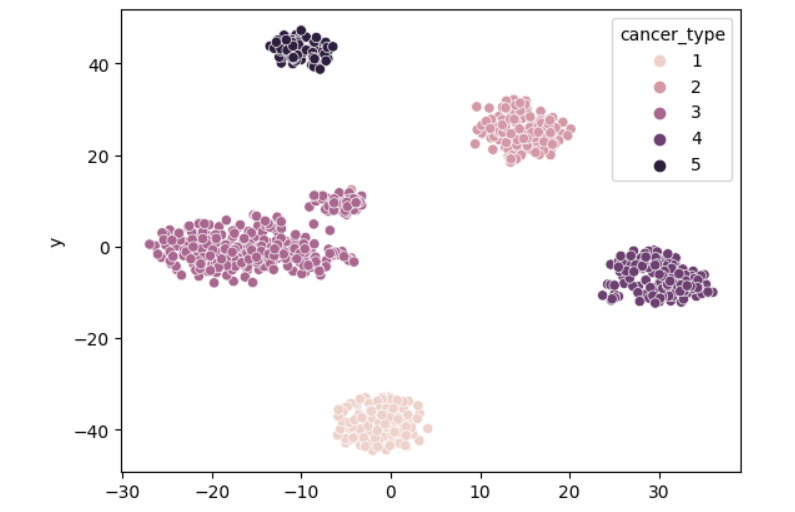
**Fig 6.4:**t-SNE Plot

Above figure shows t-SNE dimensionality reduction output as a plot. This is done so as to identify a smaller set of attributes because it is not be necessary to include all 20K gene expression values to analyze each cancer type.

df\_tsne\_data['cancer\_type']=df\_cat\_data['Class']

sns.scatterplot(x='x',y='y',hue = 'cancer\_type', data=df\_tsne\_data)

plt.show()



Above figure shows t-SNE dimensionality reduction output as a plot. This is done so as to identify a smaller set of attributes because it is not be necessary to include all 20K gene expression values to analyze each cancer type. Colour is changed from the previous t-SNE output to make it more visually understandable.

#Dimensionality reduction using LDA

df\_lda = df.drop(['Unnamed: 0'], axis=1)

df\_lda = df\_lda.drop(['Class'], axis=1)

x\_lda = df\_lda

x\_lda

x\_lda.shape

y\_lda = df['Class']

y\_lda.values

fromsklearn.discriminant\_analysis import LinearDiscriminantAnalysis as LDA

lda = LDA(n\_components=2)

x\_r2 = lda.fit(x\_lda,y\_lda).transform(x\_lda)

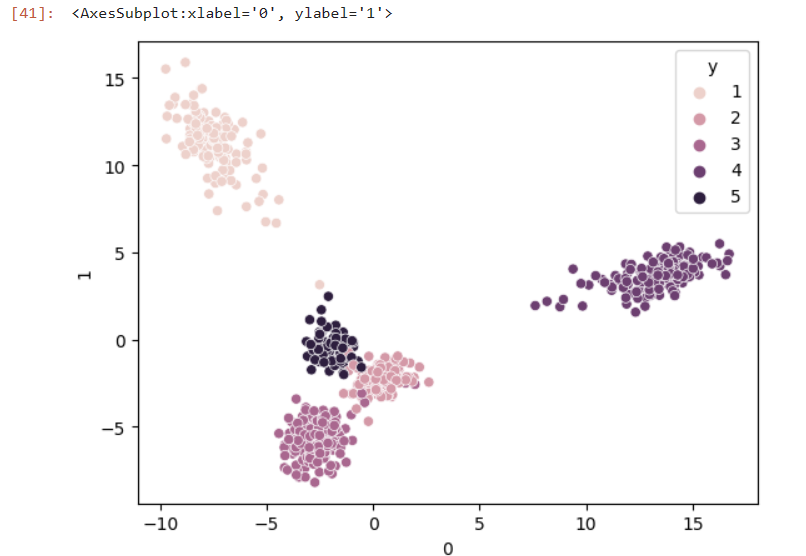
lda.explained\_variance\_ratio\_

x\_r3 = pd.DataFrame(data=x\_r2)

x\_r3['y']=y\_lda

x\_r3

sns.scatterplot(x=0,y=1,hue = 'y', data=x\_r3)



**FIG 6.5:**Dimentionality using LDA

1. **Clustering Algorithms Implementation**

This section provides the code for various clustering techniques used in the project, such as k-means, hierarchical, and mean shift clustering. It details the parameters and configurations crucial for understanding the clustering process applied to the gene expression data. So our next goal is to identify groups of genes that behave similarly across samples and identify the distribution of samples corresponding to each cancer type. Therefore, this task focuses on applying various clustering techniques, e.g., k-means, hierarchical and mean shift clustering, on genes and samples.

First, apply the given clustering technique on all genes to identify:

* Genes whose expression values are similar across all samples
* Genes whose expression values are similar across samples of each cancer type

Next, apply the given clustering technique on all samples to identify:

* Samples of the same class (cancer type) which also correspond to the same cluster
* Samples identified to be belonging to another cluster but also to the same class (cancer type)

CODE:

#Clustering Genes and Samples

#KMEANS Clustering with PCA = 2

fromsklearn.cluster import KMeans

clusters = KMeans(5, n\_init = 5)

clusters.fit(X\_pca\_with\_2)

clusters.labels\_

pca\_with\_2\_data\_frame = pd.DataFrame(data=X\_pca\_with\_2,columns=['pca1','pca2'])

pca\_with\_2\_data\_frame.head()

pca\_with\_2\_data\_frame['Cls\_label'] = clusters.labels\_

pca\_with\_2\_data\_frame['given\_cancer\_type'] = label.Class.values

pca\_with\_2\_data\_frame

brca = pca\_with\_2\_data\_frame.groupby('given\_cancer\_type').get\_group('BRCA')

brca.Cls\_label.value\_counts()

luad = pca\_with\_2\_data\_frame.groupby('given\_cancer\_type').get\_group('LUAD')

luad.Cls\_label.value\_counts()

coad = pca\_with\_2\_data\_frame.groupby('given\_cancer\_type').get\_group('COAD')

coad.Cls\_label.value\_counts()

prad = pca\_with\_2\_data\_frame.groupby('given\_cancer\_type').get\_group('PRAD')

prad.Cls\_label.value\_counts()

kirc = pca\_with\_2\_data\_frame.groupby('given\_cancer\_type').get\_group('KIRC')

kirc.Cls\_label.value\_counts()

clusters.cluster\_centers\_

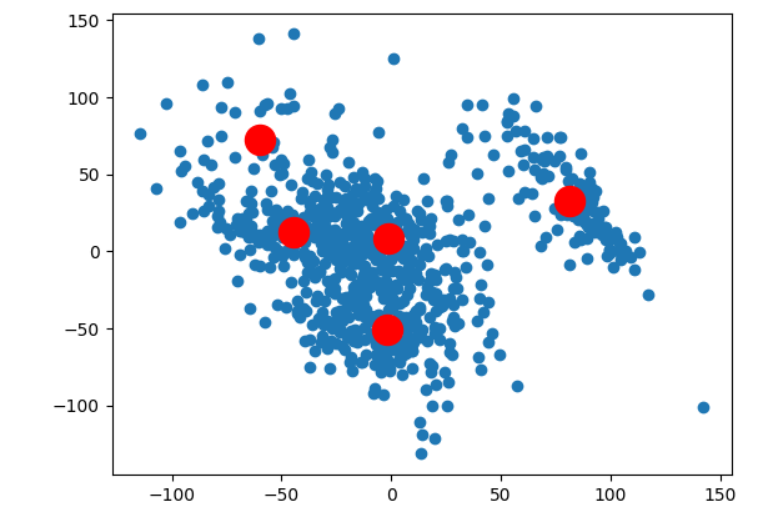
kmeans = KMeans(n\_clusters=5, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

pred\_y = kmeans.fit\_predict(X\_pca\_with\_2)

plt.scatter(X\_pca\_with\_2[:,0], X\_pca\_with\_2[:,1])

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=300, c='red')

plt.show()



**FIG 6.6:**K-means clustering with PCA

Above figure shows Kmeans clustering performed with PCA as component=2. The goal is to identify groups of genes that behave similarly across samples and identify the distribution of samples corresponding to each cancer type.

#KMEANS Clustering with PCA = .995

fromsklearn.cluster import KMeans

clusters\_995 = KMeans(5, n\_init = 5)

clusters\_995.fit(X\_pca\_with\_995)

clusters\_995.labels\_

pca\_with\_995\_data\_frame = pd.DataFrame(data=X\_pca\_with\_995)

pca\_with\_995\_data\_frame.head()

pca\_with\_995\_data\_frame['Cls\_label'] = clusters.labels\_

pca\_with\_995\_data\_frame['given\_cancer\_type'] = label.Class.values

pca\_with\_995\_data\_frame.shape

brca\_995 = pca\_with\_995\_data\_frame.groupby('given\_cancer\_type').get\_group('BRCA')

brca\_995.Cls\_label.value\_counts()

luad\_995 = pca\_with\_995\_data\_frame.groupby('given\_cancer\_type').get\_group('LUAD')

luad\_995.Cls\_label.value\_counts()

coad\_995 = pca\_with\_995\_data\_frame.groupby('given\_cancer\_type').get\_group('COAD')

coad\_995.Cls\_label.value\_counts()

prad\_995 = pca\_with\_995\_data\_frame.groupby('given\_cancer\_type').get\_group('PRAD')

prad\_995.Cls\_label.value\_counts()

kirc\_995 = pca\_with\_995\_data\_frame.groupby('given\_cancer\_type').get\_group('KIRC')

kirc\_995.Cls\_label.value\_counts()

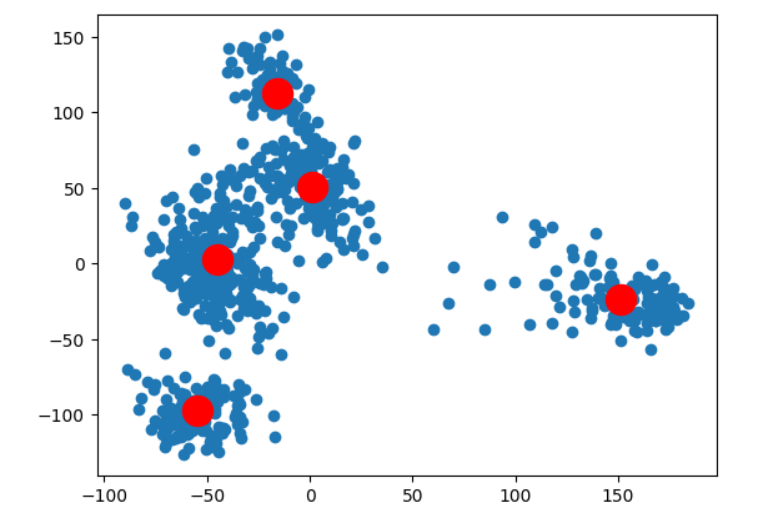
kmeans = KMeans(n\_clusters=5, init='k-means++', max\_iter=300, n\_init=10, random\_state=0)

pred\_y = kmeans.fit\_predict(X\_pca\_with\_995)

plt.scatter(X\_pca\_with\_995[:,0], X\_pca\_with\_995[:,1])

plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=300, c='red')

plt.show()



**FIG 6.7:**K-means clustering

Above figure shows Kmeans clustering performed with PCA as component=.995. The goal is to identify groups of genes that behave similarly across samples and identify the distribution of samples corresponding to each cancer type.

1. **Classification Model Building with Feature Selection**

Here, we present the full code for each machine learning model employed in this project, including Support Vector Machines (SVM), Random Forest, and Deep Neural Networks. Additionally, this section includes scripts for feature selection techniques like forward selection and backward elimination.

CODE:

#Decision tree clasifier

ml\_x = x\_lda

ml\_y = y\_lda

ml\_x.shape,ml\_y.shape

fromsklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(ml\_x,ml\_y,test\_size=0.30,random\_state=30)

fromsklearn import tree

dt\_clf = tree.DecisionTreeClassifier(max\_depth=5)

dt\_clf.fit(x\_train,y\_train)

dt\_clf.score(x\_test,y\_test)

y\_pred=(dt\_clf.predict(x\_test))

dt\_clf.score(x\_test,y\_test)

#SVM

fromsklearn.metrics import accuracy\_score

fromsklearn.svm import SVC

sv\_clf = SVC(probability=True, kernel='linear')

sv\_clf.fit(x\_train,y\_train)

sv\_clf.score(x\_test,y\_test)

y\_pred = sv\_clf.predict(x\_test)

print(accuracy\_score(y\_test,y\_pred))

#Random forest

fromsklearn import ensemble

rf\_clf = ensemble.RandomForestClassifier(n\_estimators=100)

rf\_clf.fit(x\_train,y\_train)

rf\_clf.score(x\_test,y\_test)

#Naive Bayes Classifier

fromsklearn.naive\_bayes import GaussianNB

gb\_clf = GaussianNB()

gb\_clf.fit(x\_train,y\_train)

gb\_clf.score(x\_test,y\_test)

#KNN classifier

fromsklearn.neighbors import KNeighborsClassifier

knn\_clf = KNeighborsClassifier(n\_neighbors=5)

knn\_clf.fit(x\_train,y\_train)

knn\_clf.score(x\_test,y\_test)

#Deep Neural Network

features=df.drop(['Unnamed: 0'],axis=1)

features=features.drop(['Class'],axis=1)

target=df['Class']

features.head()

target.head()

f1=features.values

y1 = pd.get\_dummies(y\_lda)

fromsklearn.model\_selection import train\_test\_split

X1\_train, X1\_valid, y1\_train, y1\_valid = train\_test\_split(f1,y1, test\_size = 0.10, random\_state=42)

X1\_train.shape,X1\_valid.shape,y1\_valid.shape,y1\_train.shape

#Define model

#SGD

importtensorflow as tf

# Initialize Sequential model

model = tf.keras.models.Sequential()

# adding layers of inout

model.add(tf.keras.layers.Dense(10000, input\_dim=20531, activation='relu', kernel\_initializer='he\_uniform'))

# Normalize the data

model.add(tf.keras.layers.BatchNormalization())

# Add 1st hidden layer

model.add(tf.keras.layers.Dense(5000, activation='relu'))

# Add 2nd hidden layer

model.add(tf.keras.layers.Dense(2000, activation='relu'))

# Add 3rd hidden layer

model.add(tf.keras.layers.Dense(1000, activation='relu'))

# Add 4th hidden layer

model.add(tf.keras.layers.Dense(500, activation='relu'))

# Add 5th hidden layer

model.add(tf.keras.layers.Dense(200, activation='relu'))

# Add 6th hidden layer

model.add(tf.keras.layers.Dense(100, activation='relu'))

# Add OUTPUT layer

model.add(tf.keras.layers.Dense(5, activation='softmax'))

# Create optimizer with non-default learning rate

sgd\_optimizer = tf.keras.optimizers.SGD(learning\_rate=0.03)

# Compile the model

model.compile(optimizer=sgd\_optimizer, loss='categorical\_crossentropy', metrics=['accuracy'])

model.summary()

history = model.fit(X1\_train,y1\_train,

validation\_data=(X1\_valid,y1\_valid),

epochs=5,

batch\_size=32)

xyz = model.predict(X1\_valid)

y\_pr=[]

for k in xyz:

y\_pr.append(np.argmax(k))

y\_val=[]

for k in y1\_valid.values:

y\_val.append(np.argmax(k))

# Confusion Matrix

fromsklearn.metrics import confusion\_matrix

confusion\_matrix(y\_val, y\_pr)

#Model evaluation

\_, train\_acc =model.evaluate(X1\_train, y1\_train, verbose=0)

\_, test\_acc = model.evaluate(X1\_valid, y1\_valid, verbose=0)

print('Train: %.3f, Test: %.3f' % (train\_acc, test\_acc))

#Plotting

plt.plot(history.history['accuracy'], label='train')

plt.plot(history.history['val\_accuracy'], label='test')

plt.xlabel('# of epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

#Apply the feature selection algorithms, forward selection and backward elimination

# automatically select the number of features for RFE

fromnumpy import mean

fromnumpy import std

fromsklearn.datasets import make\_classification

fromsklearn.model\_selection import cross\_val\_score

fromsklearn.model\_selection import RepeatedStratifiedKFold

fromsklearn.feature\_selection import RFECV

fromsklearn.tree import DecisionTreeClassifier

fromsklearn.pipeline import Pipeline

# define dataset

X, y = make\_classification(n\_samples=1000, n\_features=10, n\_informative=5, n\_redundant=5, random\_state=1)

# create pipeline

rfe = RFECV(estimator=DecisionTreeClassifier())

model = DecisionTreeClassifier()

pipeline = Pipeline(steps=[('s',rfe),('m',model)])

# evaluate model

cv = RepeatedStratifiedKFold(n\_splits=10, n\_repeats=3, random\_state=1)

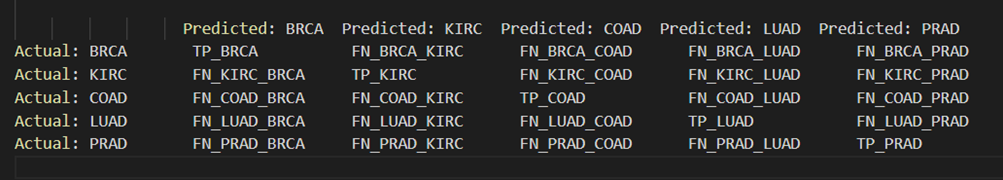
n\_scores = cross\_val\_score(pipeline, X, y, scoring='accuracy', cv=cv, n\_jobs=-1, error\_score='raise')

# report performance

print('Accuracy: %.3f (%.3f)' % (mean(n\_scores), std(n\_scores)))

**CONFUSION MATRIX**

The confusion matrix itself would look something like this for a multiclass classification problem with five cancer types (BRCA, KIRC, COAD, LUAD, and PRAD):

****

TP (True Positive) counts the cases when the cancer type was predicted correctly.

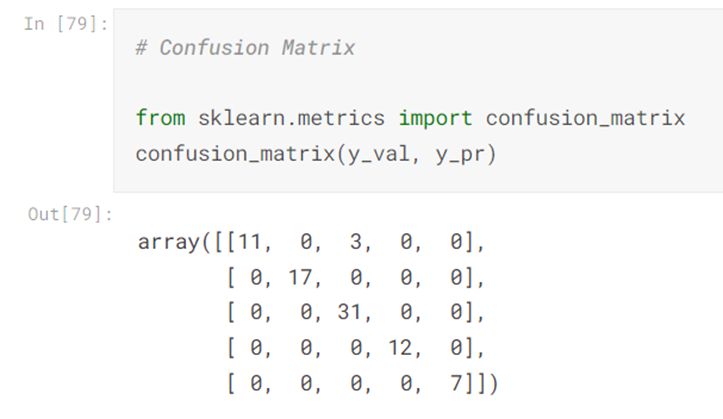
FN (False Negative) counts the cases when another cancer type was predicted (the name following FN indicates which type was incorrectly predicted instead).

For each cancer type:

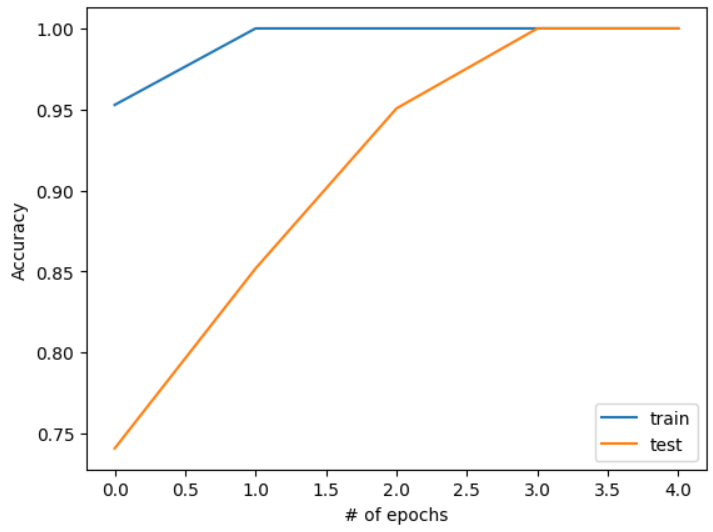
The diagonal represents the True Positives (TP) for each class.

The off-diagonal elements in each row represent the False Negatives (FN), where the model incorrectly predicted the other classes for that actual class.

Similarly, the off-diagonal elements in each column represent the False Positives (FP), where the model incorrectly predicted the actual class for other predicted classes.

****

**FIG 8.1:**Confusion matrix



**FIG 8.2:**Accuracy grapgh

The above final output image is a line graph that plots accuracy scores against the number of training epochs for a machine learning model. Two curves are depicted: one for the training set (in blue) and one for the test set (in orange). The x-axis represents the number of epochs, which range from 0 to 4, and the y-axis represents the accuracy, ranging from 0.75 to 1.00 (or 75% to 100%).

The test curve starts at an accuracy of around 0.78 and increases steadily as the number of epochs increases, ending just below 0.90. The upward trend of the test curve indicates that the model is learning and generalizing well to new data. The consistent improvement over time suggests that additional epochs are beneficial in increasing the model's performance on the test set.

It's important to note that the accuracy on the test set is slightly lower than the training set, which is expected since the model was trained on the training set. The goal is to have the test curve as close as possible to the training curve, which would indicate good generalization. However, if the test curve starts to decrease while the training curve continues to increase, it would be a sign of overfitting

.

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

The project focused on analyzing gene expression data across various types of cancers, such as breast, renal, colon, lung, and prostate cancer, represents a significant advancement in oncological research. Starting with a comprehensive amalgamation and examination of datasets containing expression values of over 20,000 genes from 802 samples, the project laid a robust groundwork for delving into the complex world of cancer genomics. This initial phase, involving the creation of hierarchically-clustered heatmaps and null-hypothesis testing, provided crucial insights into the dataset's structure and patterns, setting the stage for deeper analysis.

In the subsequent phase, dimensionality reduction techniques like PCA, LDA, and t-SNE were adeptly employed, skillfully condensing the extensive genomic data into a more manageable and informative form. This was crucial for focusing on the most relevant attributes for analyzing different cancer types. The project then progressed to the clustering of genes and samples using techniques such as k-means, hierarchical, and mean shift clustering. This crucial phase grouped genes with similar expression patterns and elucidated the distribution of samples across various cancer types, providing valuable insights into the genetic underpinnings of these diseases.

The final and perhaps most pivotal phase of the project was the development of sophisticated classification models using multiclass SVM, Random Forest, and Deep Neural Networks. The thoughtful integration of feature selection algorithms further honed the model's accuracy, and the validation of selected genes through statistical significance testing added a layer of scientific rigor to the findings. These efforts culminated in a comprehensive understanding of cancer genomics, contributing significantly to the literature and paving the way for future research endeavors.

Looking ahead, there are several promising avenues for enhancing this foundational work. Expanding the dataset to include a broader array of cancer types, stages, and demographic backgrounds will enhance the generalizability and depth of the findings. Integrating additional genomic data types, such as whole-genome sequencing or epigenetic markers, could provide a more holistic understanding of cancer genetics. The application of advanced machine learning and deep learning techniques, including exploring unsupervised and semi-supervised learning methods, could offer improved model accuracy and novel insights. Adding a longitudinal analysis component would allow tracking gene expression over time, shedding light on cancer progression and treatment efficacy.

Furthermore, collaborating with clinical researchers to validate and apply these findings in real-world healthcare settings could lead to the development of novel diagnostic tools and personalized treatment plans. Conducting clinical trials based on the project's insights could have a tangible impact on cancer treatment and patient care. Finally, expanding the project to investigate the interplay between environmental, lifestyle, and genetic factors in cancer development could offer a comprehensive view of the disease, opening doors to preventative strategies and public health interventions.

In conclusion, this project represents a substantial contribution to the field of cancer genomics, blending meticulous data analysis with innovative machine learning techniques. Its findings not only enhance our understanding of cancer but also offer a framework for future research, potentially leading to groundbreaking advancements in cancer diagnosis and treatment. The project's methodical approach, combined with its significant findings, justifies its recognition as a notable academic endeavor, one that holds the promise of influencing future oncological research and patient care.

**REFERENCES**

[1] Aradhita Bhandari, B. K. Tripathy, Khurram Jawad, Surbhi Bhatia, Mohammad Khalid Imam Rahmani, and ArwaMashat “Cancer Detection and Prediction Using Genetic Algorithms” published on 16th may,2022

[2] Deidra Jordan and DeEtta Mills “Past, Present, and Future of DNA Typing for Analyzing Human and Non-Human Forensic Samples” published on March 22 ,2021

[3] SourabhKatoch&Sumit Singh Chauhan & Vijay Kumar “A review on genetic algorithm: past, present, and future” published on October 31 ,2020

[4] Francis S. Collins, M.D., Ph.D., Jennifer A. Doudna, Ph.D., Eric S. Lander, Ph.D., and Charles N. Rotimi, Ph.D. “Human Molecular Genetics and Genomics — Important Advances and Exciting Possibilities” ” published on January 7 ,2021

[5] Benjamin Hunter 1,2 , SumeetHindocha 1,3 and Richard W. Lee “The Role of Artificial Intelligence in Early Cancer Diagnosis” published on March 16 ,2022

[6] Joshua D. Schiffman, MD, Paul G. Fisher, MD, and Peter Gibbs, MBBS, FRACP, MD “Early Detection of Cancer: Past, Present, and Future” published on March 16 ,2022

[7] FadiAlharbi and AleksandarVakanski, “Machine Learning Methods for Cancer Classification Using Gene Expression Data” published on Jan 28, 2023.

[8] David G. P. van I Jzendoorn, KarolySzuhai, Inge H. Briaire-de Bruijn, Marie Kostine, Marieke L. Kuijjer “Machine learning analysis of gene expression data reveals novel diagnostic and prognostic biomarkers and identifies therapeutic targets for soft tissue sarcomas” published on Feb 20 ,2019.

# [9] Rebecca C. Fitzgerald, Antonis C. Antoniou, LjiljanaFruk&Nitzan Rosenfeld “The future of early cancer detection” published on April 19 ,2022

# [10] F.J. Shaikh, D.S. Rao “Prediction of Cancer Disease using Machine learning Approach” published on April 16 ,2021

# [11] Xingdong Chen, Jeffrey Gole, Athurva Gore, Qiye He, Ming Lu, Jun Min, Ziyu Yuan, Xiaorong Yang, Yanfeng Jiang, Tiejun Zhang, Chen Suo, Xiaojie Li, Lei Cheng, Zhenhua Zhang, HongyuNiu, Zhe Li, Zhen Xie, Han Shi, Xiang Zhang, Min Fan, Xiaofeng Wang, Yajun Yang, Justin Dang, Catie McConnell, Juan Zhang, Jiucun Wang, Shunzhang Yu, Weimin Ye, Yuan Gao, Kun Zhang, Rui Liu & Li Jin “Non-invasive early detection of cancer four years before conventional diagnosis using a blood test” published on July 21 ,2020

[12] Erik Karger and Marko Kureljusic "Artificial Intelligence for Cancer Detection" published on Jan 29,2023.

[13] Ibrahim M. Nasser and Samy S. Abu-Naser "Lung Cancer Detection Using Artificial Neural Network" published on Mar 3,2019.

[14] kavitha singh,Usha Chauhan and LokeshVarshney "Impact of covid-19 in lung cancer detection using image processing techniques, artificial intelligence and machine learning approaches" published on Aug 11,2023.

[15] Mustafa Mohammed Jassim and Mustafa Musa Jaber "Systematic review for lung cancer detection and lung nodule classification: Taxonomy, challenges, and recommendation future works" Published on Aug 10,2022.