Introduction:

Cancer genomics is a pivotal field in genetic biology and biomedicine. Understanding cancer's origins guides treatment plans and drug development. High-throughput gene sequencing's cost reduction has driven personalised cancer treatment. Predicting cancer incidence through gene sequencing is crucial for early diagnosis and treatment. I analysed data from 802 people with different cancer types, each with 20K+ gene expression values. The goal of the project is to create a robust classification model.

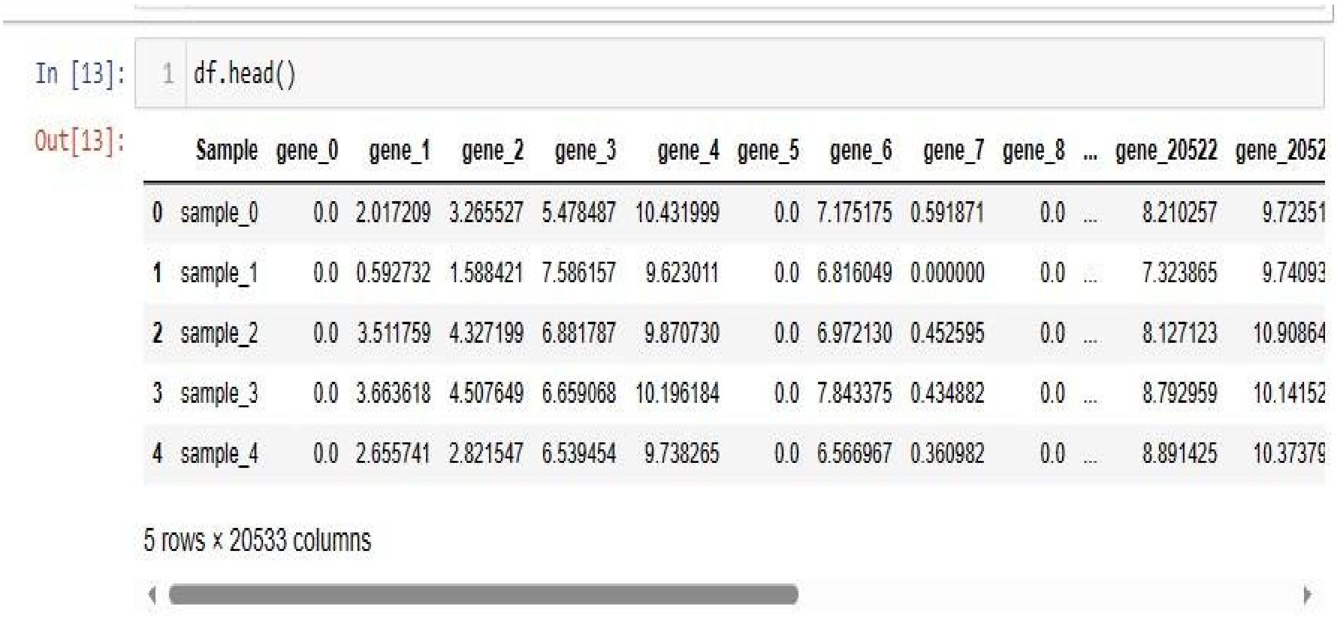
Methodology Used:

1. Exploratory Data Analysis:

Merge both the datasets. Plot the merged dataset as a hierarchically-clustered heatmap. Perform Null-hypothesis testing.

The data set comprises two components: patient numbers and their corresponding cancer types. In total, there are 801 patients with five different cancer types. The second component includes cancer genome data for each patient, containing 20K genes. My initial step involves merging these two data sets to create a consolidated data set

```
df = pd.merge(df_1, df_2, on ="Unnamed: 0")
```



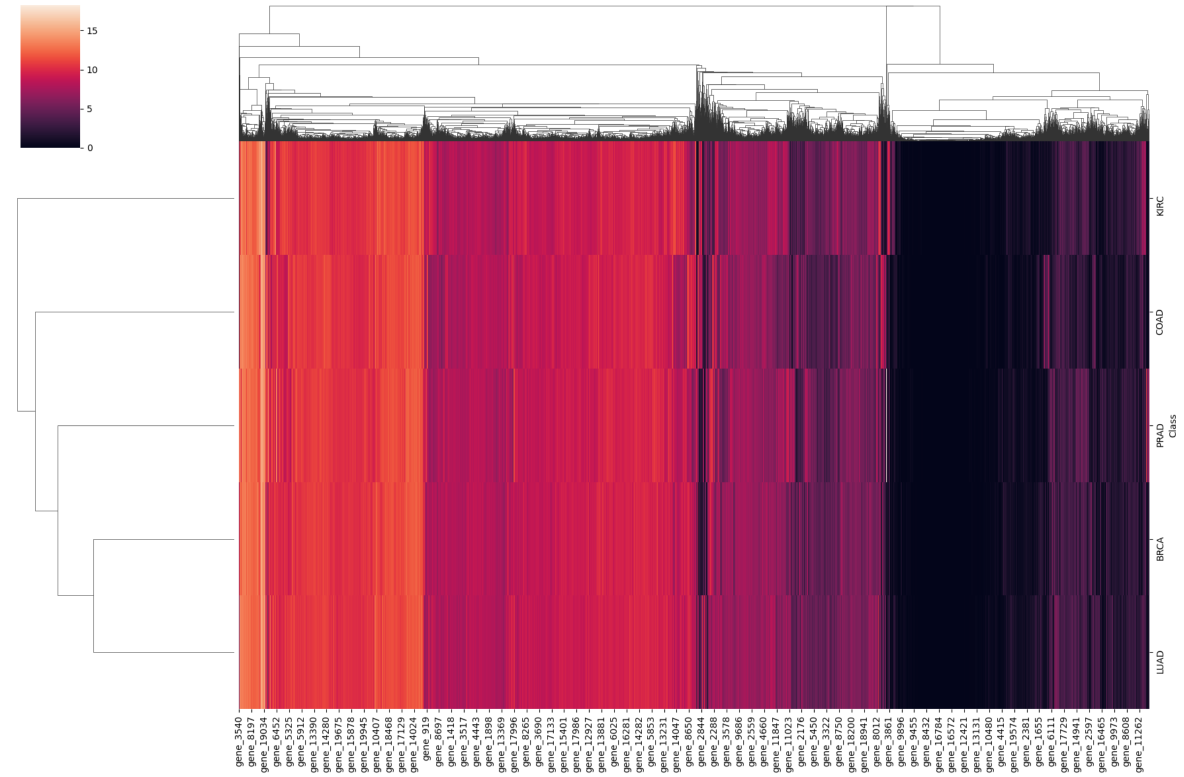
```
In [13]: df.head()
```

Out[13]:

	Sample	gene_0	gene_1	gene_2	gene_3	gene_4	gene_5	gene_6	gene_7	gene_8	...	gene_20522	gene_2052
0	sample_0	0.0	2.017209	3.265527	5.478487	10.431999	0.0	7.175175	0.591871	0.0	...	8.210257	9.72351
1	sample_1	0.0	0.592732	1.588421	7.586157	9.623011	0.0	6.816049	0.000000	0.0	...	7.323865	9.74093
2	sample_2	0.0	3.511759	4.327199	6.881787	9.870730	0.0	6.972130	0.452595	0.0	...	8.127123	10.90864
3	sample_3	0.0	3.663618	4.507649	6.659068	10.196184	0.0	7.843375	0.434882	0.0	...	8.792959	10.14152
4	sample_4	0.0	2.655741	2.821547	6.539454	9.738265	0.0	6.566967	0.360982	0.0	...	8.891425	10.37379

5 rows × 20533 columns

Heatmap:



2. Dimensionality Reduction Method:

Each sample has expression values for around 20K genes. However, it may not be necessary to include all 20K genes expression values to analyse 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 dimensionality reduction using various techniques such as, PCA, LDA, and t-SNE. Input: Complete dataset including all genes (20531) Output: Selected Genes from each dimensionality reduction method

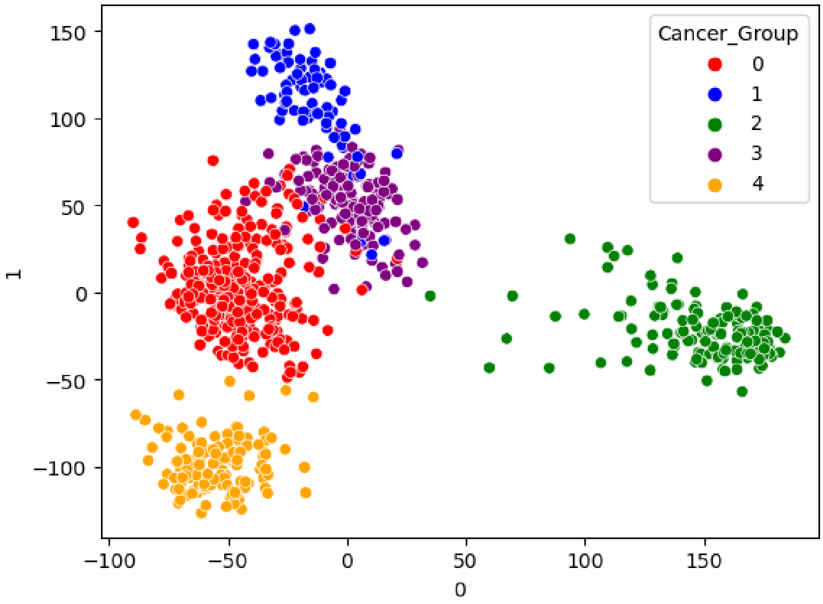
In the context of this project, each sample is associated with expression values for approximately 20,000 genes. However, it's often unnecessary to include all 20,000 gene expression values for the analysis of each cancer type. Therefore, the primary focus of this project revolves around implementing dimensionality reduction techniques, specifically PCA, LDA, and t-SNE, to identify a more manageable set of attributes for building multiclass classification models.

2.1 PCA:

Principal Component Analysis (PCA) is a widely used dimensionality reduction technique and statistical method in the field of data analysis and machine learning. Its primary purpose is to reduce the complexity of high-dimensional data while preserving as much of the relevant information as possible.

Fig 1:

custom_palette = {0: 'red', 1: 'blue', 2: 'green', 3: 'purple', 4: 'orange'}
sns.scatterplot(x=0,y=1, hue = 'Cancer_Group',data=df_995, palette=custom_palette)

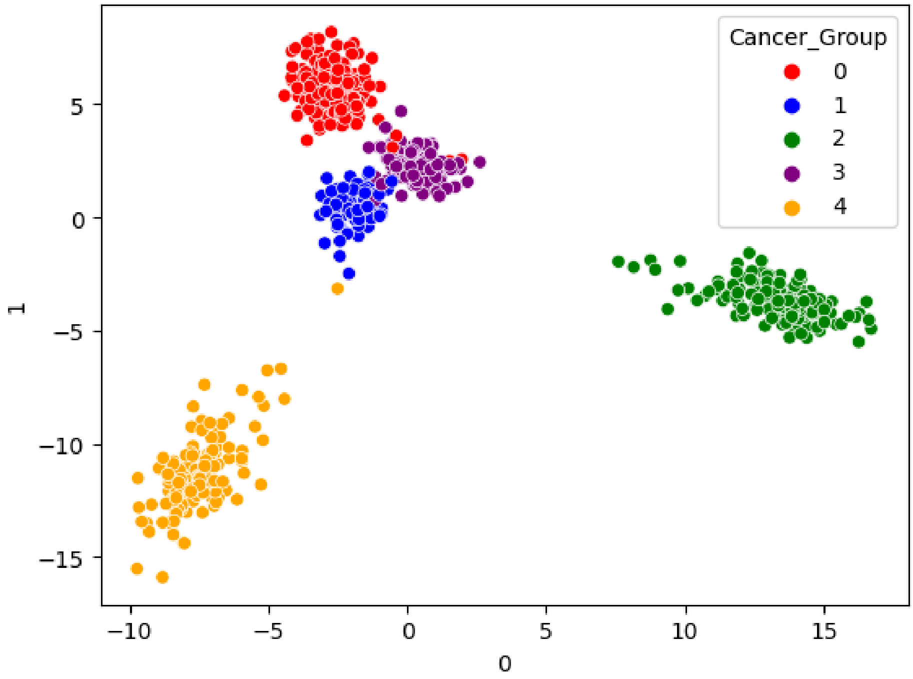


2.2 LDA:

LDA (Linear Discriminant Analysis) is distinct from PCA, which focuses on maximising variance. LDA aims to project data into a lower-dimensional space, where data of the same type cluster closely while data of different types spread apart .

Fig:1

custom_palette = {0: 'red', 1: 'blue', 2: 'green', 3: 'purple', 4: 'orange'}
sns.scatterplot(x=0,y=1, hue =Cancer_Group',data=data_lda, palette=custom_palette

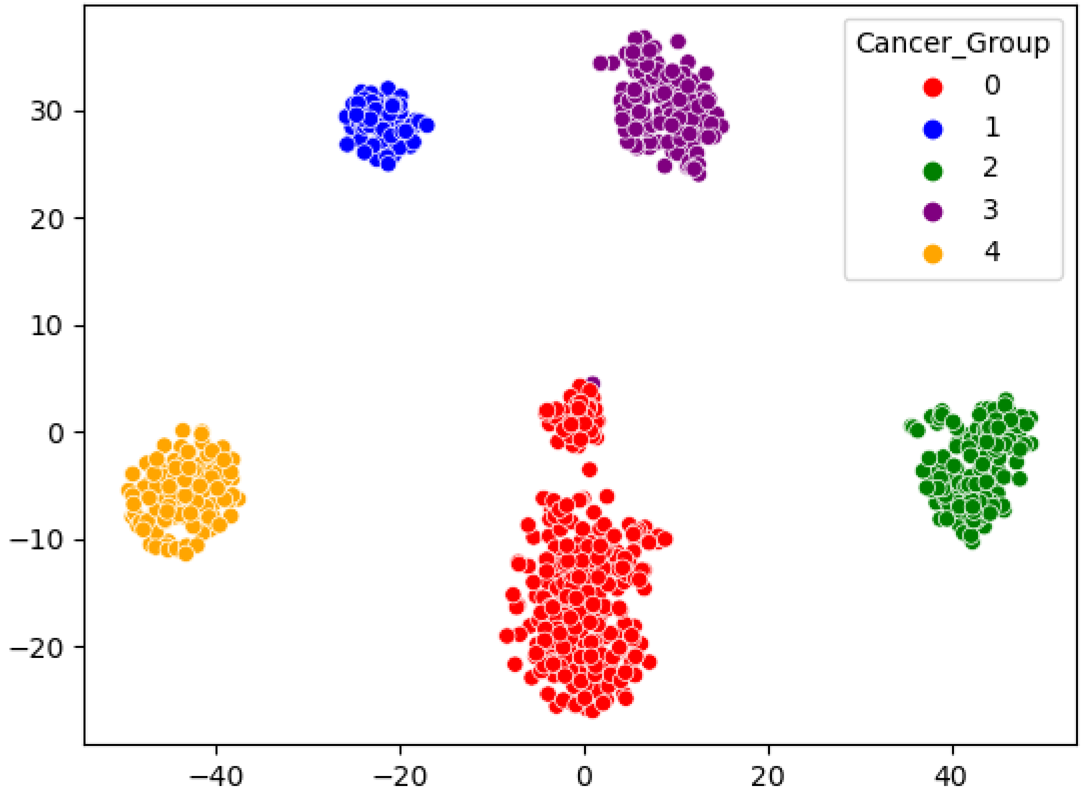


2.3 t-SNE dimensionality reduction

T-Distributed Stochastic Neighbour Embedding (t-SNE) primarily serves as a powerful tool for visualising and navigating high-dimensional data. Its primary objective is to transform a multi-dimensional dataset into a lower-dimensional representation. To achieve this, t-SNE starts by selecting a random data point and calculating the Euclidean distances to all other data points. Data points in close proximity to the selected one yield higher similarity values, while those farther away yield lower similarity values. These similarity values are then used to construct a similarity matrix for each data point.

Fig1

```
custom_palette = {0: 'red', 1: 'blue', 2: 'green', 3: 'purple', 4: 'orange'}
sns.scatterplot(x=ts_data_features[:,0],y=ts_data_features[:,1], hue = 'Cancer_Group',data=ts_data,
palette=custom_palette)
```



3. Clustering Genes and Samples:

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:

1. Genes whose expression values are similar across all samples
2. Genes whose expression values are similar across samples of each cancer type

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

1.Samples of the same class (cancer type) which also correspond to the same cluster

2.Samples identified to be belonging to another cluster but also to the same class (cancer type)

3.1 K-Means:

K-means clustering is an unsupervised machine learning technique that divides a dataset into distinct clusters based on similarity, with "k" specifying the desired number of clusters. It involves iterative steps to assign data points to clusters and update cluster centroids until convergence, revealing underlying patterns or groupings in the data.

Fig:1

```
k_means = KMeans(n_clusters=5, init="k-means++", max_iter=500, n_init=10, random_state=42)
pred_y = k_means.fit_predict(X_pca_995)
sns.scatterplot(x=X_pca_995[:,0],y=X_pca_995[:,1],palette=custom_palette)
sns.scatterplot(x=k_means.cluster_centers_[:,0],y=k_means.cluster_centers_[:,1],s=300, c="red")
```

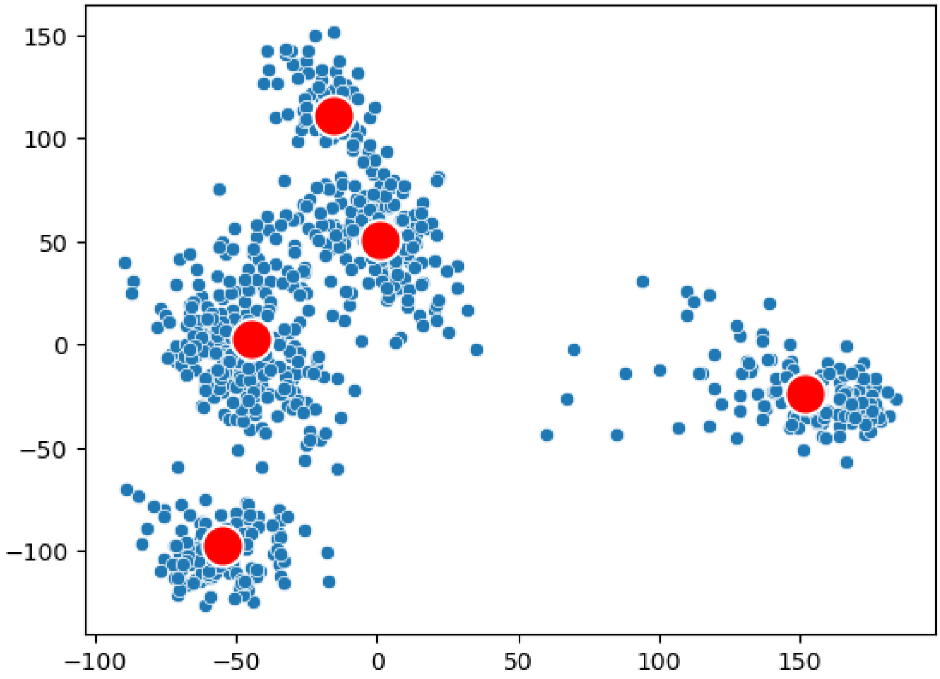
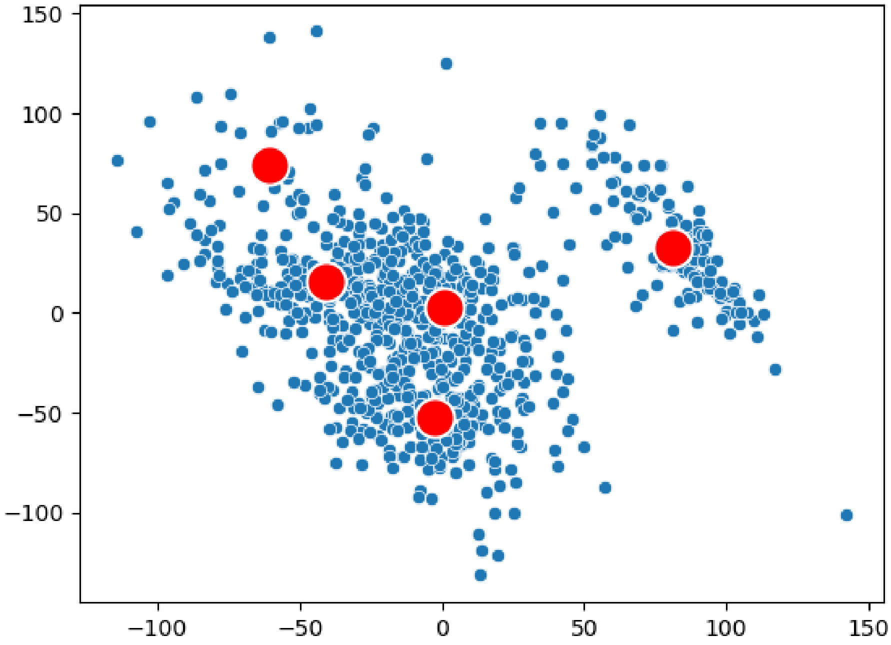


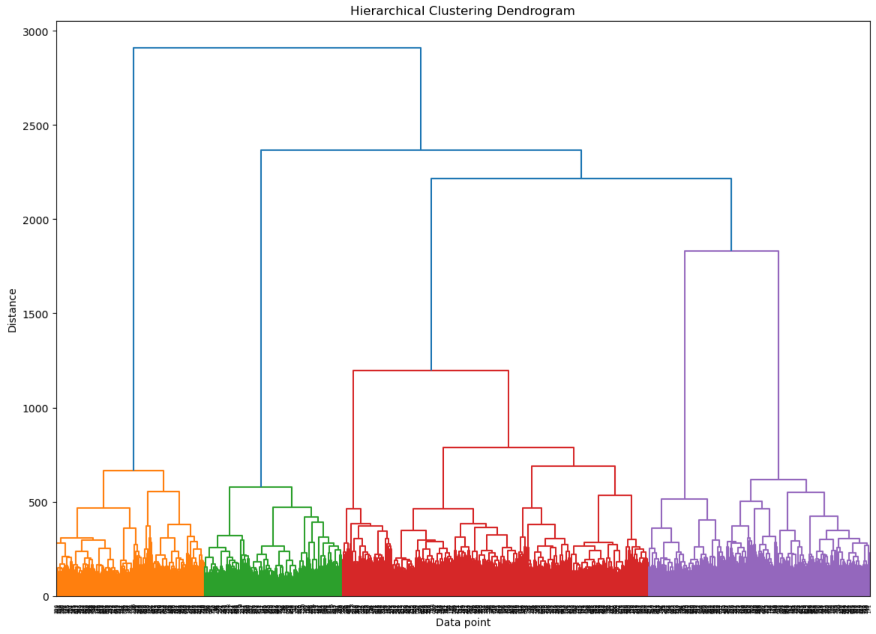
Fig2:

```
k_means_2 = KMeans(n_clusters=5, init="k-means++", max_iter=500, n_init=10, random_state=42)
pred_y = k_means_2.fit_predict(X_pca_2)
sns.scatterplot(x=X_pca_2[:,0],y=X_pca_2[:,1])
sns.scatterplot(x=k_means_2.cluster_centers_[:,0],y=k_means_2.cluster_centers_[:,1],s=300, c="red")
```

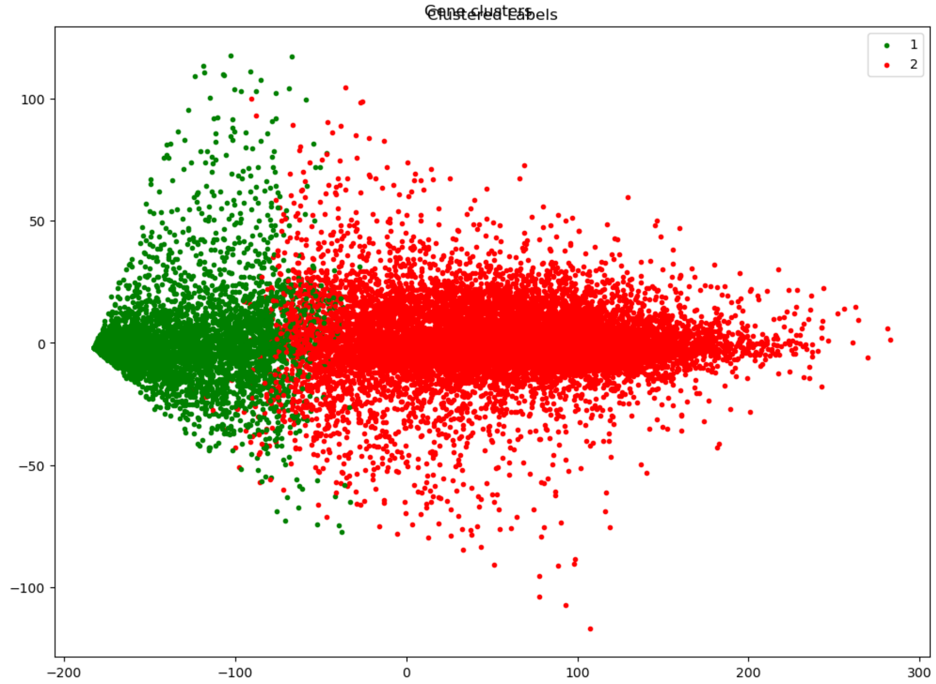


3.3 Hierarchical Clustering

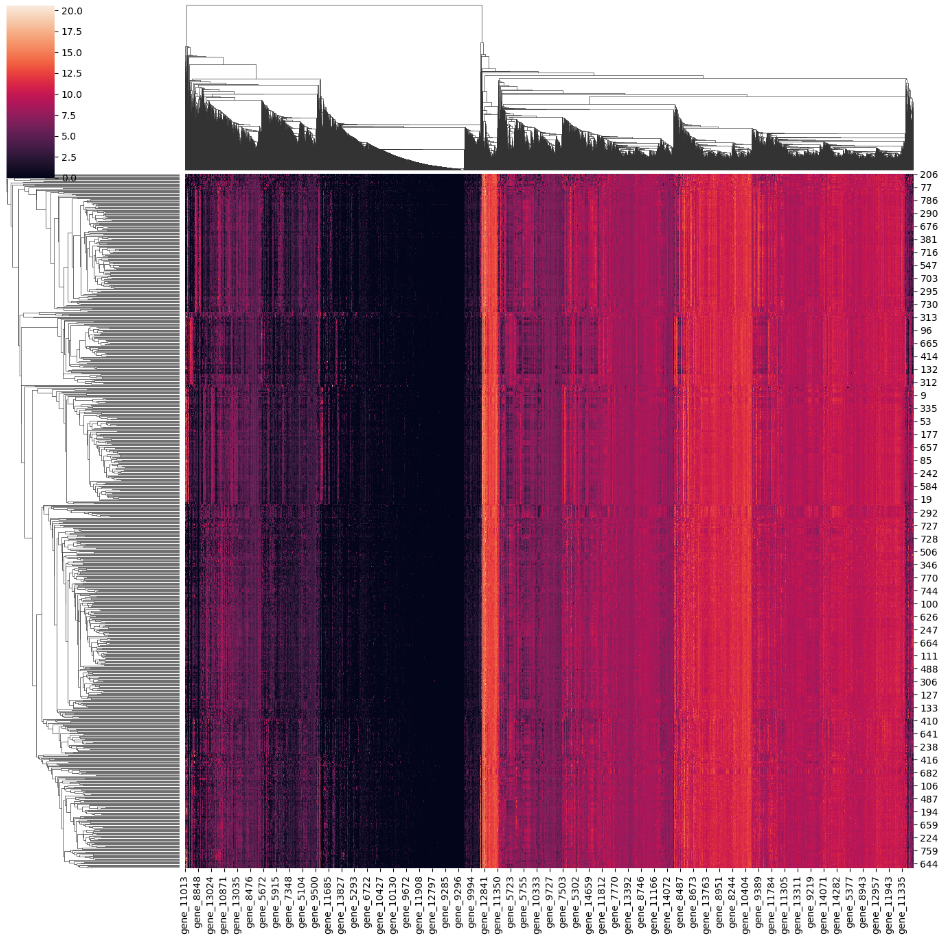
The linkage matrix contains the history of which data points were merged into which cluster during each iteration. If n is no.of samples,there are n-1 rows. Each row in the matrix has four columns. The first two columns are either data points or cluster labels that are being merged, the third column is the cluster distance between the first two column values, and the last column is the total number of data points in the cluster once the merge is complete.



Visualising Gene Clusters:



Clustered Heatmap:



4. Building Classification Model(s) with Feature Selection:

Our final task is to build a robust classification model(s) for identifying each type of cancer.

Sub-tasks:

Build a classification model(s) using multiclass SVM, Random Forest, and Deep Neural Network to classify the input data into five cancer types

Apply the feature selection algorithms, forward selection, and backward elimination to refine selected attributes (selected in Task-2) using the classification model from the previous step

Validate the genes selected from the last step using statistical significance testing (t-test for one vs. all and F-test)

The objective is to discover sets of genes exhibiting similar behaviour across samples and pinpoint the sample distributions associated with each cancer type. The ultimate aim is to construct robust classification models capable of identifying each cancer type. I will be building classification models using multiclass SVM, Random Forest, and Deep Neural Network .

4.1 Decision Tree Classifiers:

Decision tree classifiers are a machine learning tool used for making decisions or predictions by following a tree-like structure. They are easy to understand and interpret, making them valuable in various applications. One common use case is in healthcare to predict patient outcomes, like diagnosing diseases or recommending treatments based on a patient's medical history and symptoms. Decision tree classifiers are also used in finance for credit scoring to determine creditworthiness, and in marketing for customer segmentation to target specific groups with tailored promotions.
The result we got using the Decision tree classifier with max depth 5 is 1.0.

4.2 SVM:

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. Its primary objective is to find a hyperplane that best separates data into distinct categories. SVM is widely used in various applications, such as image classification, text classification, and bioinformatics. It is valued for its ability to handle high-dimensional data and its robustness in dealing with complex decision boundaries.

The result of validation accuracy we got using SVM: 0.9984

4.3 Random Forest Classifier:

Random Forest is a type of ensemble learning, part of the Bagging method, that combines the strength of multiple decision trees. Each tree in the Random Forest is built on a random subset of data and features. By taking a vote from these trees (majority vote for classification or averaging for regression), it delivers robust and accurate predictions. This method is particularly useful in tasks like classification and regression.

We got the accuracy score of 0.996875 using Random Forest Classifiers

4.4 Naive Bayes Classifier

The Naive Bayes Classifier is a group of straightforward probabilistic classifiers based on Bayes' theorem, making the "naive" assumption that features are independent of each other. It assigns class labels to instances based on their features. While it comprises multiple algorithms, they all adhere to the fundamental idea that features are treated as uncorrelated with each other.

We got the accuracy score of 0.9875 using Naive Bayes Classifier

4.5 KNN:

K-Nearest Neighbors (KNN) is a simple and intuitive machine learning algorithm used to classify data points based on their similarity to nearby data points. In your scenario, KNN could be applied to predict cancer types. It would work by examining the gene expression values of a new sample and identifying the "k" nearest samples in the dataset with known cancer types. The majority cancer type among these nearest neighbours would be assigned as the predicted cancer type for the new sample. KNN is particularly useful when there is no clear separation between cancer types and when local patterns in gene expression data are essential for classification.

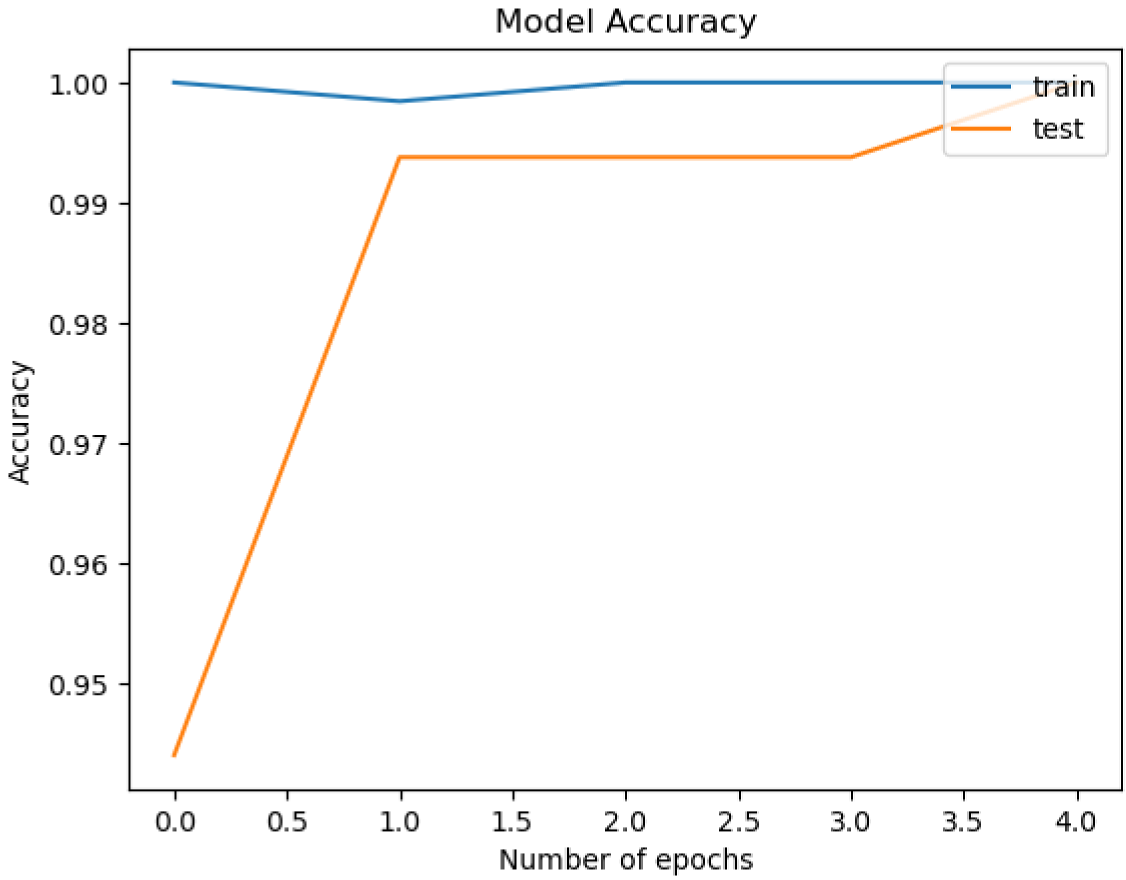
We got the accuracy score of 1 using Naive Bayes Classifier

4.6 Deep Neural Network:

The Deep Neural Network (DNN) is a key component in our project, designed for deep learning tasks. It uses a neural network with hidden layers to model complex data patterns.

In our project, DNN is trained using the backpropagation algorithm to improve its predictive accuracy, making it suitable for tasks like cancer type prediction.

We optimise DNN with Stochastic Gradient Descent (SGD), an efficient method that accelerates training. Although it may slightly reduce convergence speed, it greatly reduces computational load in high-dimensional problems .



Using Neural Network with Stochastic Gradient Descent, we achieved a high accuracy score of 1.0 in our project, demonstrating its effectiveness in our classification task.

5. Conclusion:

Within the scope of our project, we delved into a gene dataset associated with diverse cancer types. Our primary focus was to condense the dataset's complexity, which we accomplished through Principal Component Analysis (PCA). In the subsequent phase, we engaged in unsupervised learning, using clustering techniques to discern underlying data patterns.

In the pursuit of classifying these cancer types, we harnessed various classification models, including Support Vector Machine (SVM), Random Forest, and Deep Neural Network. Among these models, SVM emerged as the champion, delivering an impressive accuracy rate of 100%. This resounding success underscores the efficacy of our model in effectively distinguishing between different cancer types.