**Ovarian Cancer Dataset's Analysis**

# Team 18: Aditya Kanala, Maruti Mohit Rayalacheruvu

# **Introduction**

## ***Background***

Ovarian cancer is a type of cancer that originates in the ovaries, the reproductive organs in women responsible for producing eggs and hormones. It is a significant health concern, with various cell types actively dividing in the ovaries, leading to the development of tumors.

This form of cancer is challenging to detect in its preliminary stages due to a lack of specific symptoms. As a result, it often goes unnoticed until it reaches an advanced stage, making effective treatment more complex.

Epithelial ovarian cancer is the most common type, and risk factors include age, family history, and inherited gene mutations like BRCA1 and BRCA2. Symptoms may include abdominal bloating, pelvic pain, difficulty eating, and frequent urination.

Treatment typically involves surgical removal of the tumor, and in some cases, the ovaries, and surrounding tissues. Chemotherapy and targeted therapy may also be recommended to eliminate remaining cancer cells.

The prognosis for ovarian cancer depends on factors such as the stage at diagnosis, the type of cancer, and the individual's overall health. Early detection is crucial for better outcomes, but currently, there are challenges in developing effective screening methods.

Research efforts are ongoing to enhance our understanding of ovarian cancer, improve early detection methods, and develop more effective treatments. Increased awareness and support for ovarian cancer research are essential in addressing this complex and often late-diagnosed disease.

## ***Motivations***

Choosing ovarian cancer as a data science project presents a compelling opportunity to make a meaningful impact on healthcare. Several motivations drive the selection of this project:

1. **Enhancing Early Detection:** The core motivation centers around addressing the challenge of late-stage diagnoses in ovarian cancer. Data science techniques enable the exploration of diverse datasets, empowering the development of models that can identify subtle patterns indicative of early-stage ovarian tumors. Improving early detection is a key step toward enhancing patient outcomes.
2. **Comprehensive Dataset Exploration:** The project recognizes the richness of available data, encompassing patient demographics, genetic information, imaging, and clinical records. Motivated by a desire for a holistic understanding, data science methodologies will be applied to explore relationships and patterns across these varied data types, contributing to a comprehensive view of ovarian tumors.
3. **Machine Learning for Diagnosis:** Machine learning algorithms have shown promise in diagnosing diseases by identifying patterns in large datasets. Applying these algorithms to ovarian cancer data can enhance diagnostic accuracy and assist healthcare professionals in making informed decisions.
4. **Informed Decision-Making:** The motivation extends beyond model development to the practical application of insights in clinical settings. By unraveling complex relationships within the data, the project aims to provide healthcare professionals with actionable information, facilitating more informed decision-making regarding patient care strategies and treatment plans.
5. **Advancing Scientific Knowledge:** Ovarian cancer remains a challenging research area, and a data-driven approach can advance our scientific understanding. Insights gained from analyzing large-scale datasets may uncover novel correlations, driving further research and contributing to the collective knowledge of ovarian cancer.

In summary, choosing ovarian cancer as a data science project aligns with the broader goals of improving early detection, advancing personalized medicine, and contributing to the overall fight against a challenging and often late-diagnosed disease.

## ***Goal***

The overarching goal of this data science initiative is to revolutionize the landscape of ovarian cancer diagnosis and treatment by leveraging advanced data analysis, machine learning, and bioinformatics. The main objective is to develop sophisticated predictive models and data-driven tools capable of accurately detecting ovarian cancer. These models will focus on a comprehensive analysis of gene composition, including nuanced aspects such as gene expression levels and gene nomenclature.

## ***Specific Objectives:***

**Innovative Methodologies:** This project aims to pioneer innovative methodologies in data analysis, machine learning, and bioinformatics. By pushing the boundaries of current practices, the goal is to introduce innovative techniques that can extract meaningful insights from complex genomic data.

**Robust, Evidence-Based Models:** The development of robust and evidence-based predictive models is a central objective. These models will be constructed meticulously, incorporating a diverse range of genetic information to ensure accuracy, reliability, and generalizability across diverse patient populations.

**Early Detection and Precision:** The primary focus is on early detection of ovarian cancer by identifying subtle patterns and distinctive gene signatures. The goal is to pinpoint indicators of ovarian tumors at their nascent stages, allowing for more timely and precise diagnoses. This, in turn, contributes to refining the categorization of ovarian tumors, facilitating more efficacious and personalized therapeutic interventions.

**Enhanced Patient Outcomes:** The ultimate aim is to significantly enhance patient outcomes. Achieving more timely and accurate diagnoses empowers healthcare professionals to implement interventions at an earlier stage, potentially improving treatment efficacy and patient survival rates.

In summary, this project sets out to change the landscape of ovarian cancer care by advancing the capabilities of data science and bioinformatics. Through the development of sophisticated models, the goal is to transform the diagnosis and treatment of ovarian cancer, leading to improved patient outcomes and a more personalized approach to healthcare.

# **Methodology**

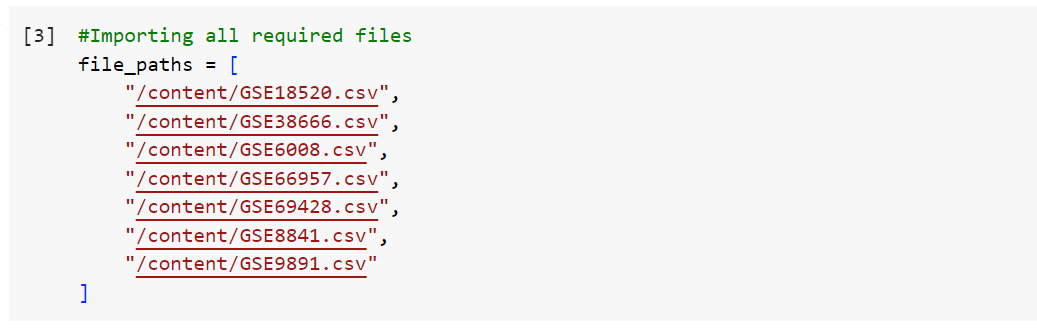
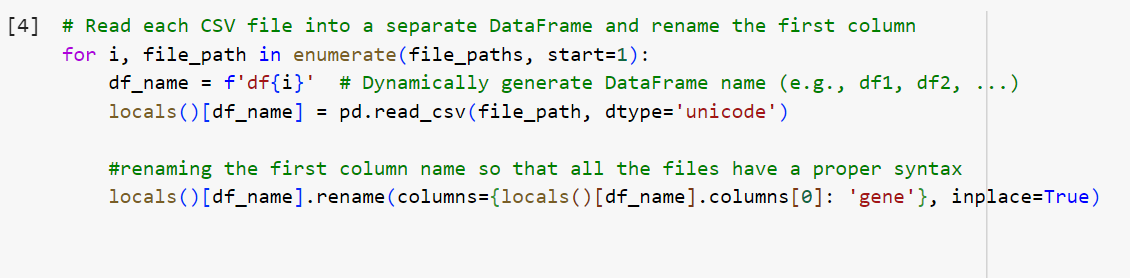
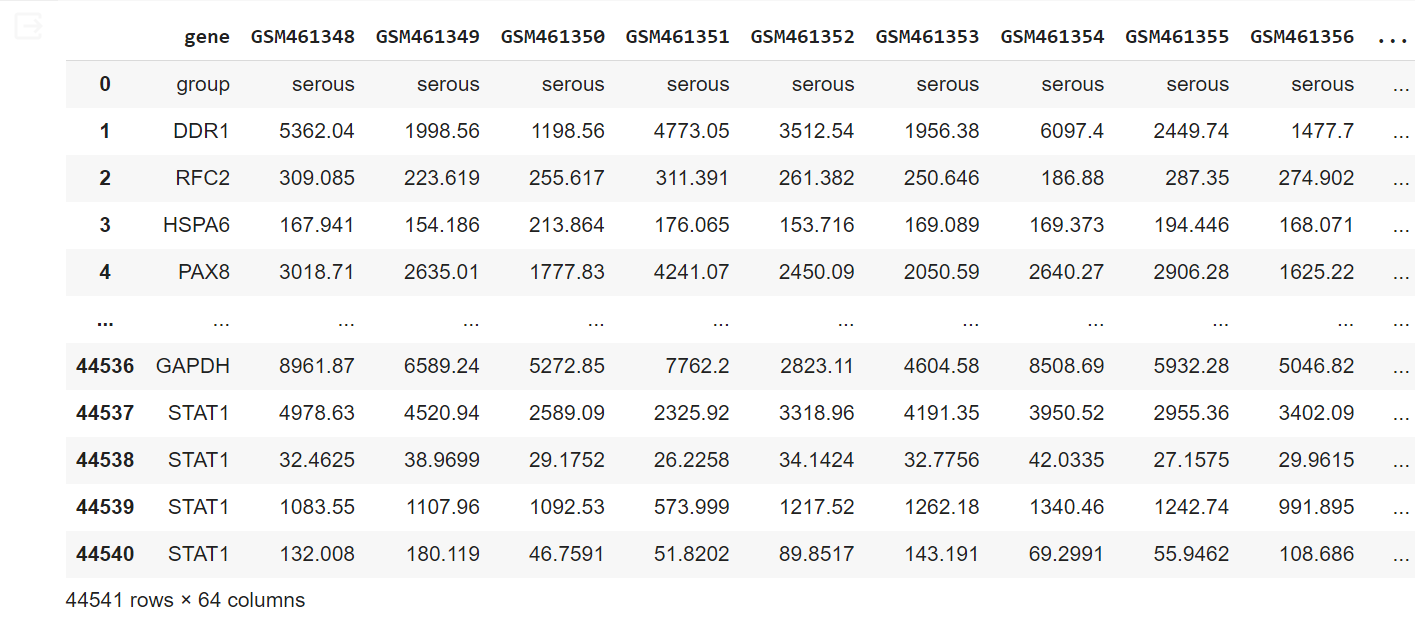
[The link for the project notebook can be found here](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0?usp=sharing)

Every side heading redirects to its specific part of the code which helps users to find the required code instead of searching the whole notebook

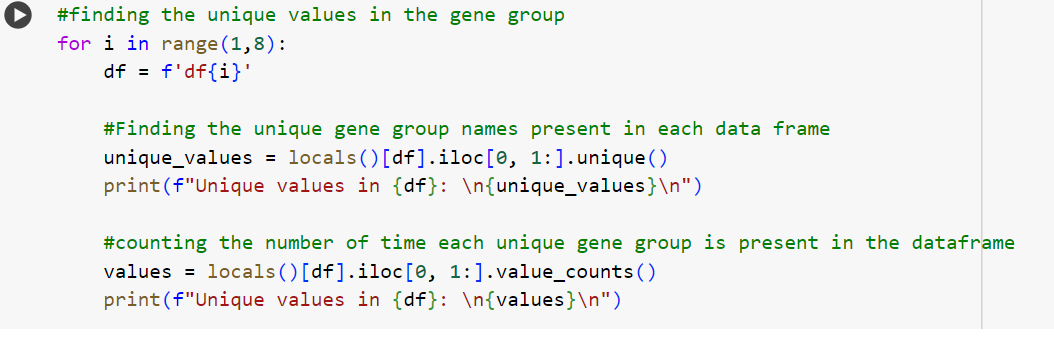
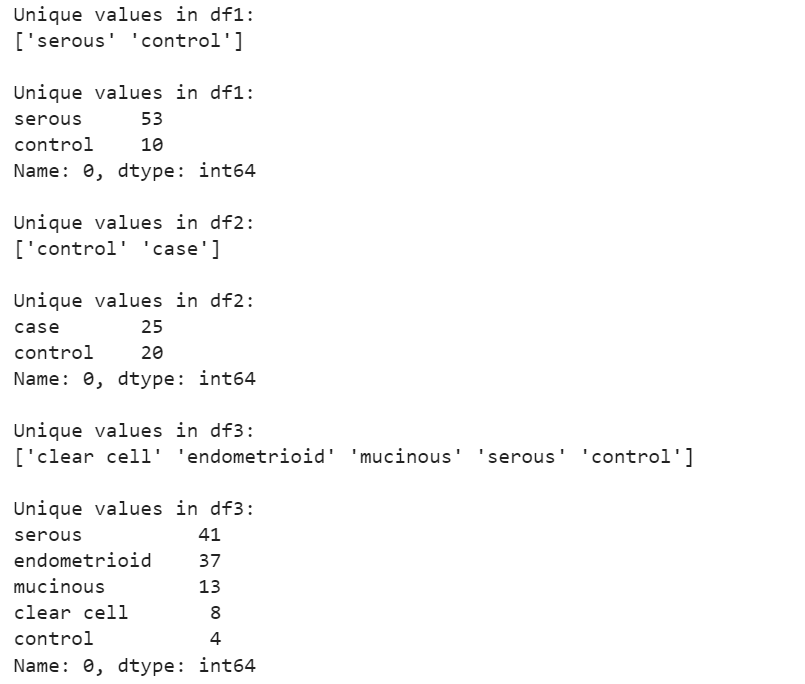
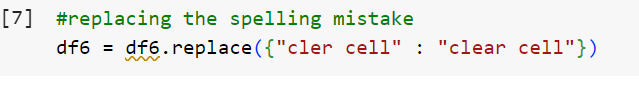
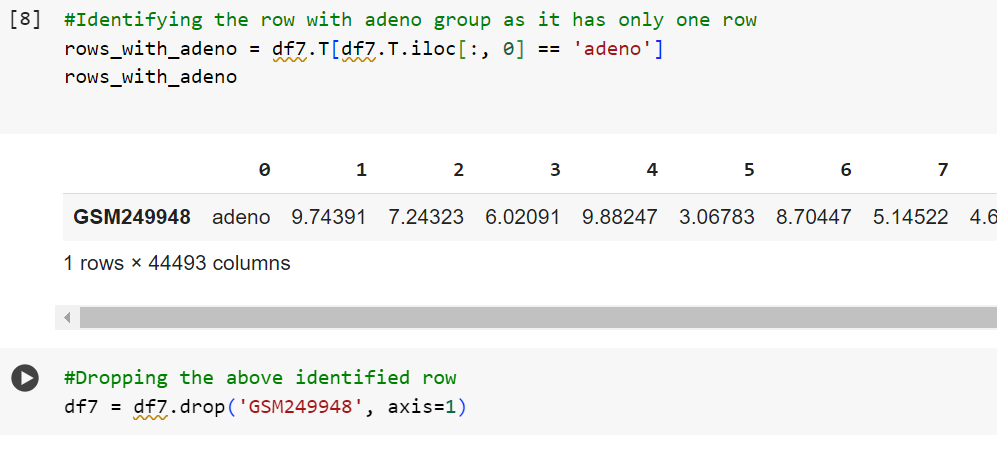
The methodology consists of the below major steps

1. ***Data Collection*** 
   1. In the data collection phase, the main objective is to gather relevant datasets essential for the analysis or problem at hand. This involves acquiring data from various sources, which can include databases, APIs, online repositories, or other means. The goal is to assemble a comprehensive and representative data set aligning with the project's objectives.
2. ***Exploratory Data Analysis (EDA)***
   1. EDA is a crucial step where the collected data is analyzed and visualized to gain insights and understand the underlying patterns. During EDA, various statistical and graphical methods are employed to explore the characteristics of the dataset. This includes summary statistics, distribution plots, correlation analysis, and data visualization techniques. EDA helps identify trends, outliers, and potential relationships between variables, laying the foundation for informed decision-making.
3. ***Feature Engineering and Preprocessing***
   1. Feature engineering involves transforming or creating new features from the existing ones to enhance the predictive power of the model. This step includes handling missing data, encoding categorical variables, scaling numerical features, and creating interaction terms. Feature preprocessing ensures that the data is in a suitable format for modeling. Techniques such as normalization, standardization, and handling outliers are applied to prepare the dataset for the machine learning algorithms.
4. ***Model Building and Evaluation***
   1. In the decisive step, machine learning models are developed based on the preprocessed dataset. This involves selecting appropriate algorithms that align with the nature of the problem, such as regression for predicting continuous outcomes or classification for predicting categories. Models are trained on a subset of the data and evaluated using another subset to assess their performance. Evaluation metrics, such as accuracy, precision, recall, F1-score, or regression metrics, provide insights into how well the model generalizes to unseen data. Model tuning and optimization may be performed to enhance performance.

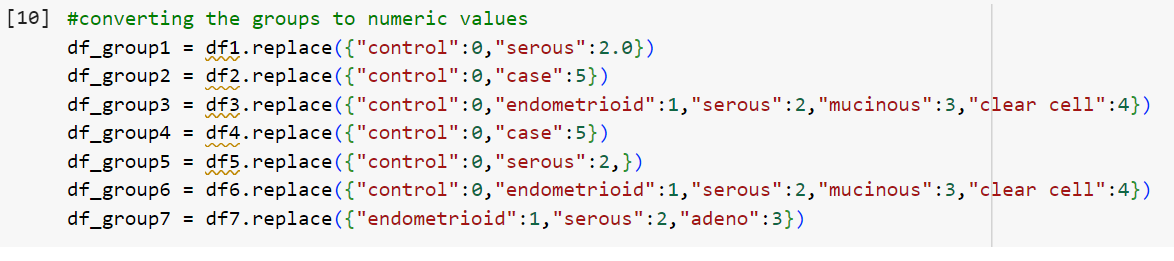
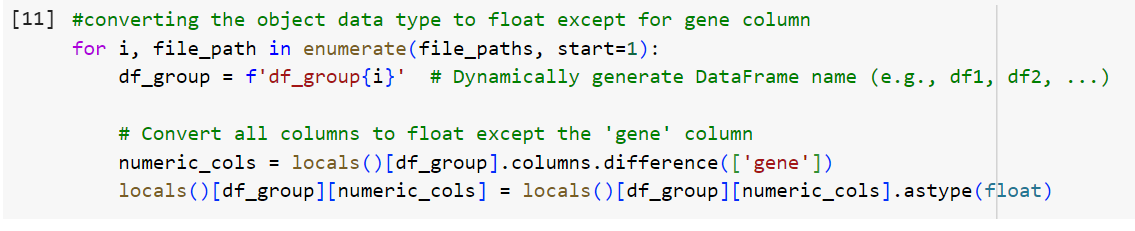
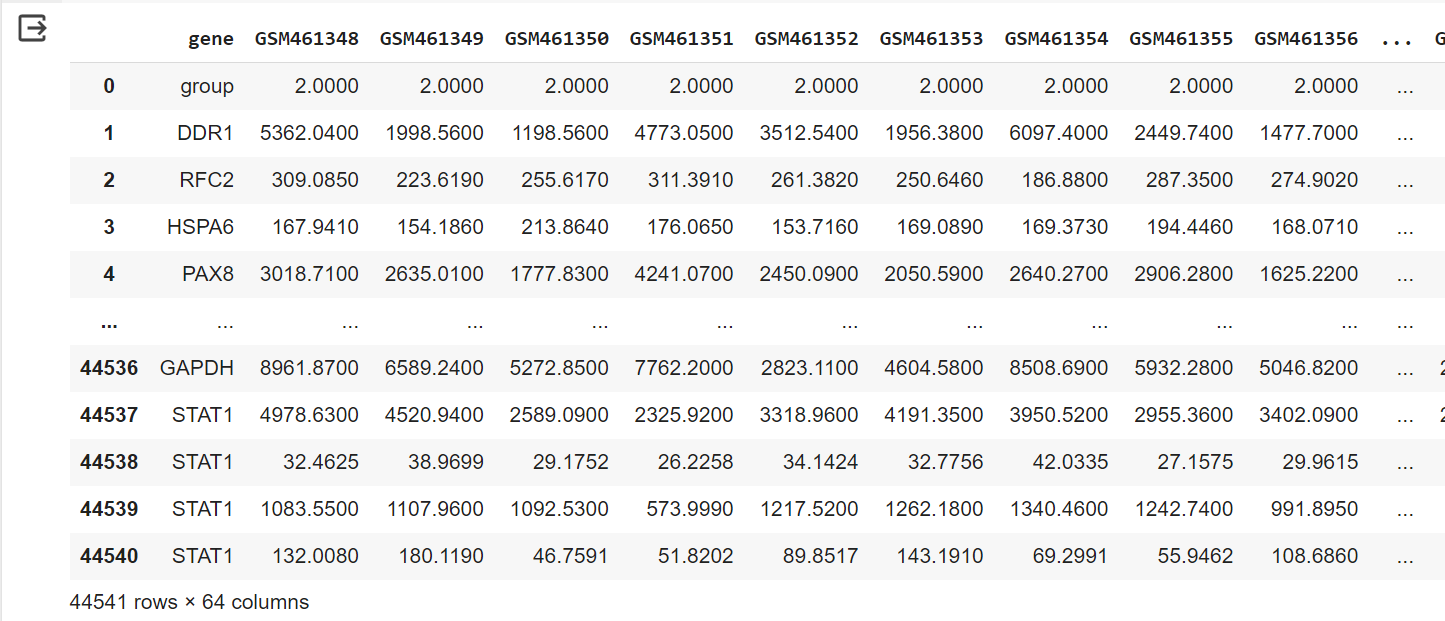
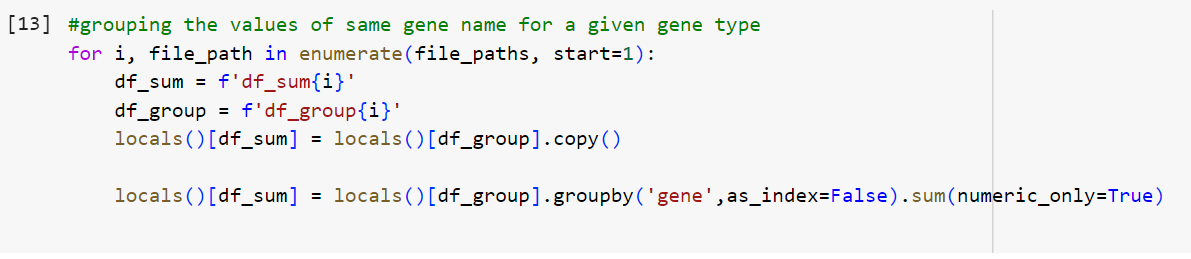
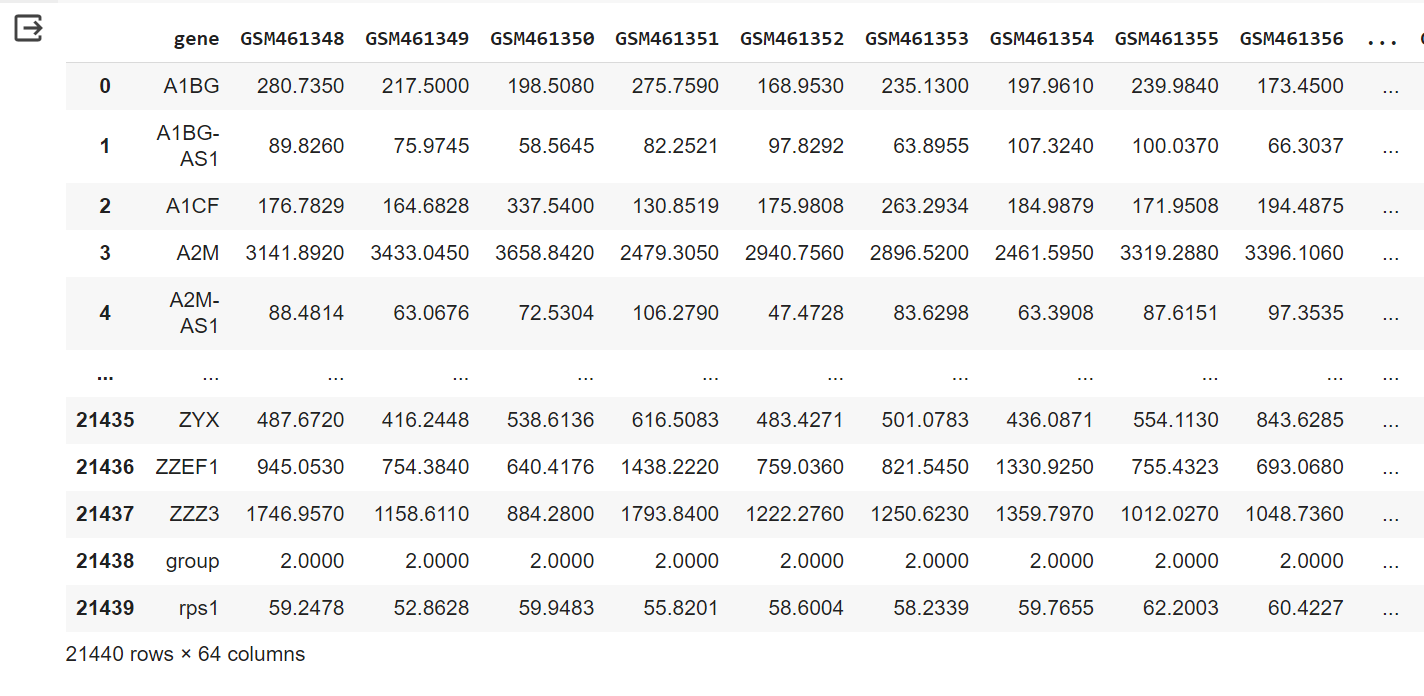
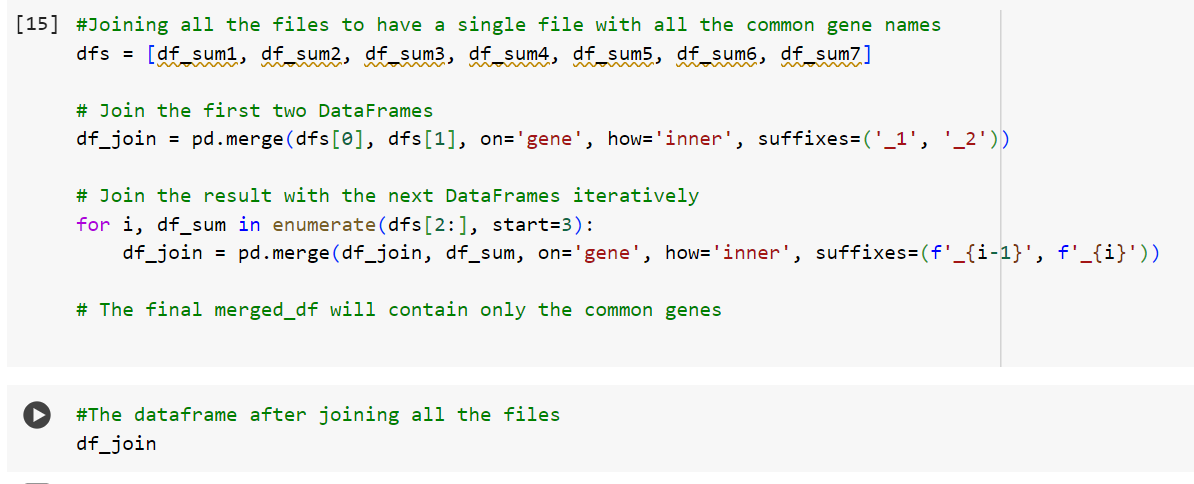
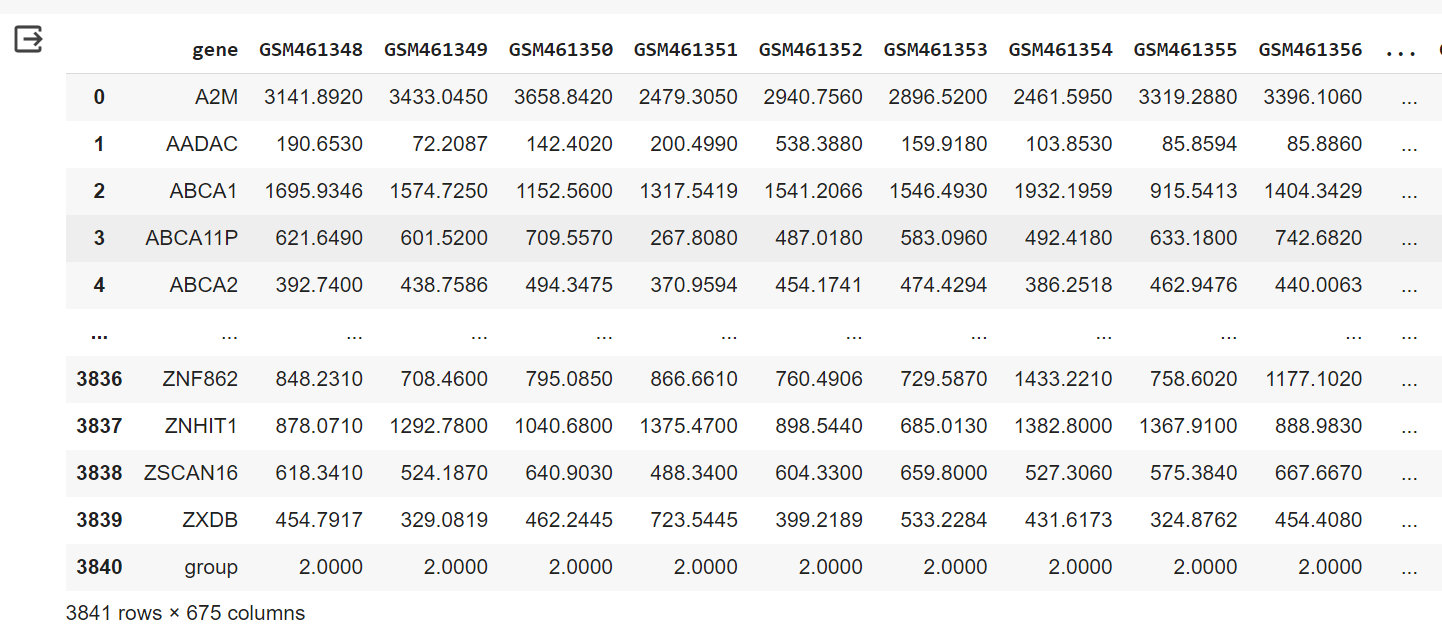
## ***STEP 1 - [Data Collection](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=4vwyD2JLfEFj)***

* Used the Ovarian Cancer Data from the Kaggle database.
  + [Ovarian Cancer Dataset](https://www.kaggle.com/datasets/yoshifumimiya/6-ovarian-cancer-datasets/?select=GSE18520.csv)
* Loaded multiple CSV files using pandas, each representing various gene types along with the associated gene name with their respective expression levels for different cancer groups.
* Gathered gene expression data from distinct files and stored in an array.
  + 
* Dynamically named DataFrames for each file, ensuring standardized syntax by renaming the first column to 'gene.'
  + 
  + 

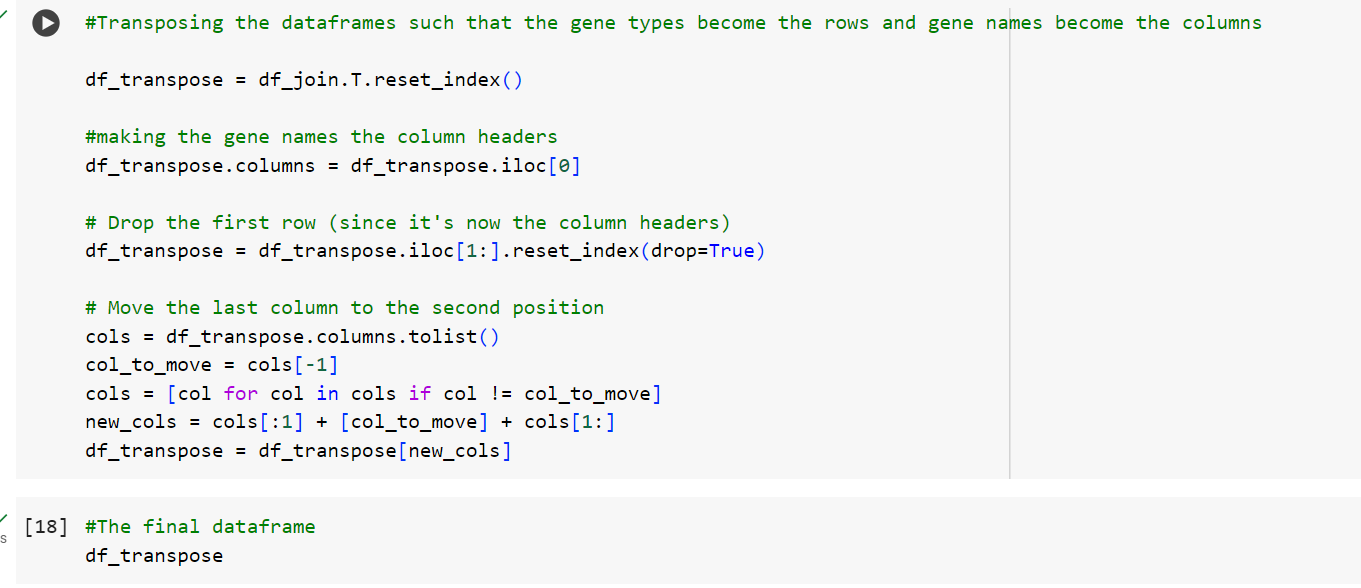
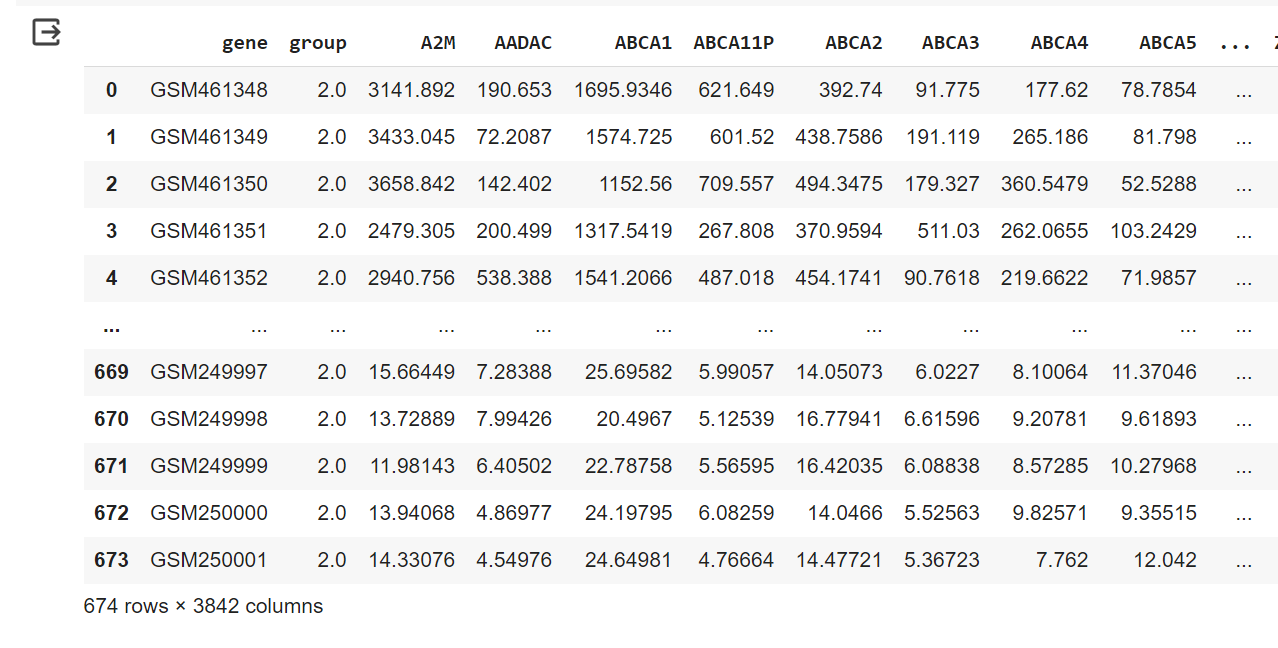
## ***STEP 2 - [Exploratory Data Analysis (EDA)](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=hx3teLOKgDuk)***

* EDA conducted to understand the data better.
* Explored unique gene group values in each DataFrame, providing insights into the diversity of gene types.
  + 
  + 
* Detected and rectified anomalies, such as a spelling mistake in one dataset ('cler cell' to 'clear cell').
  + 
* Identified and dropped a row related to the 'adeno' group, ensuring data integrity.
  + 

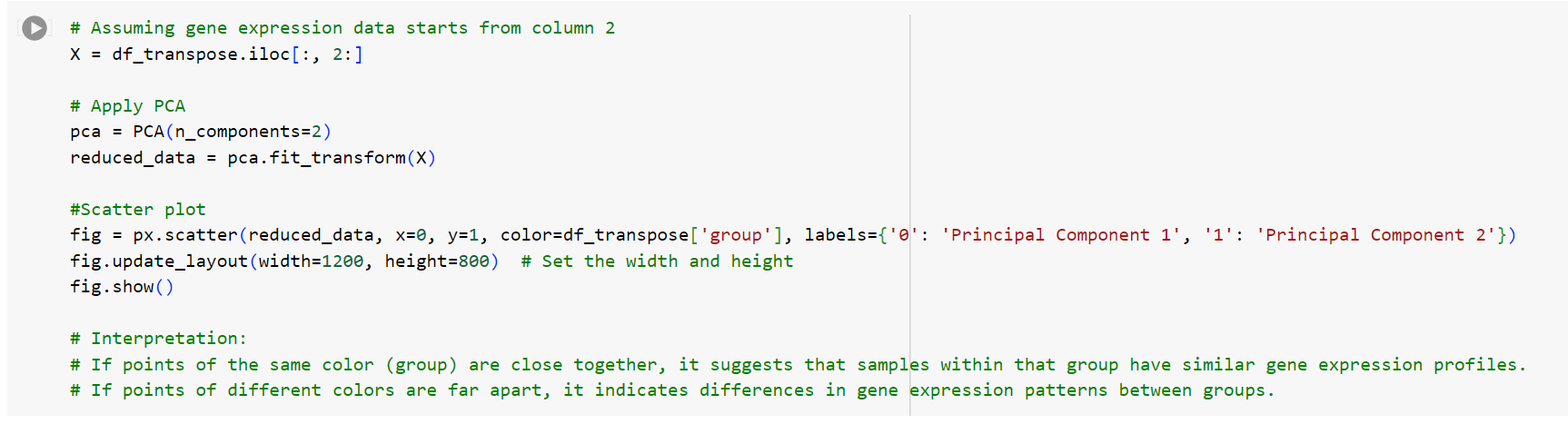
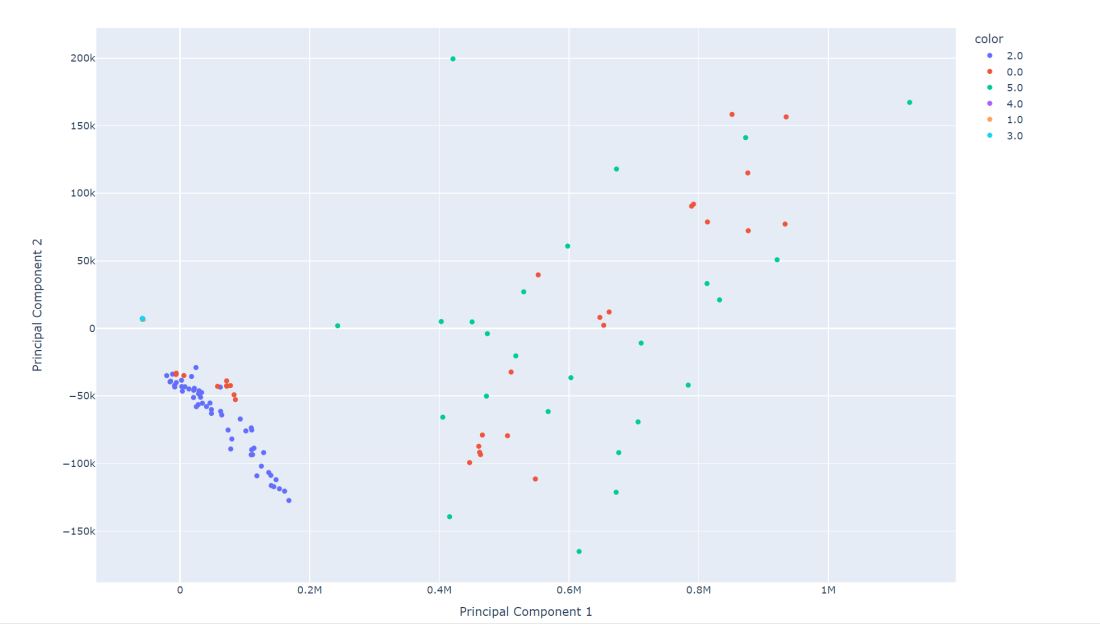
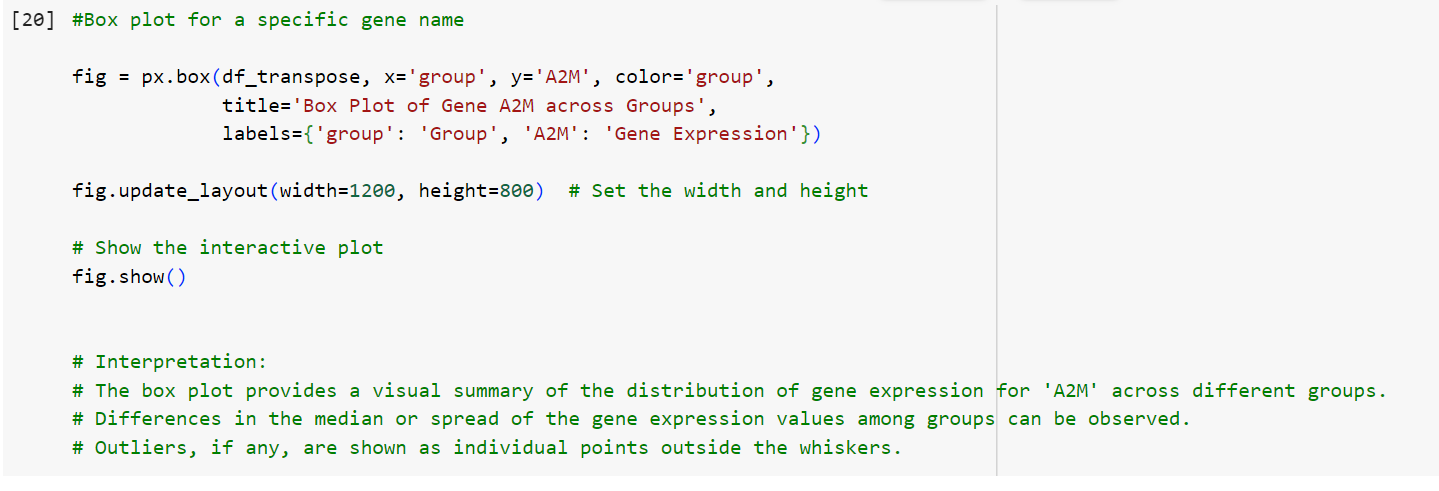
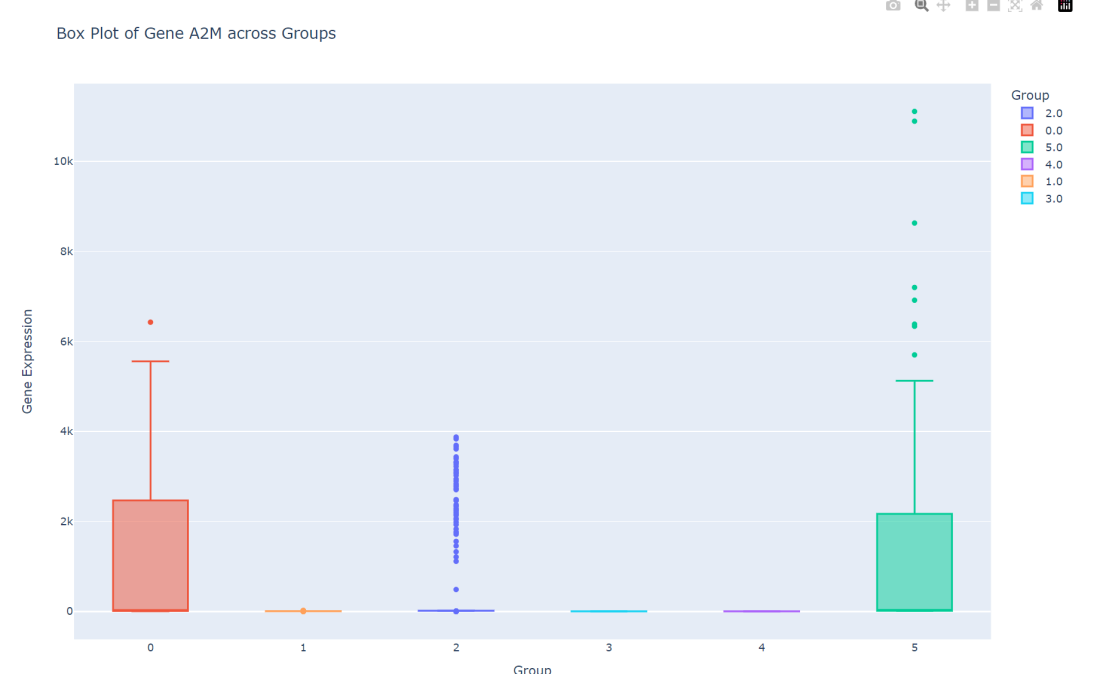
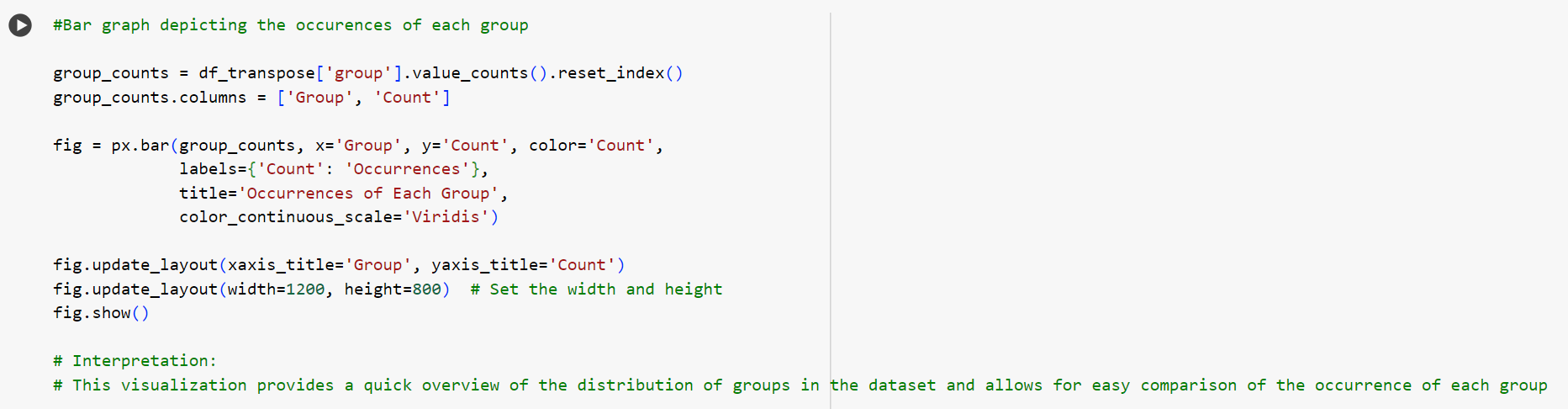
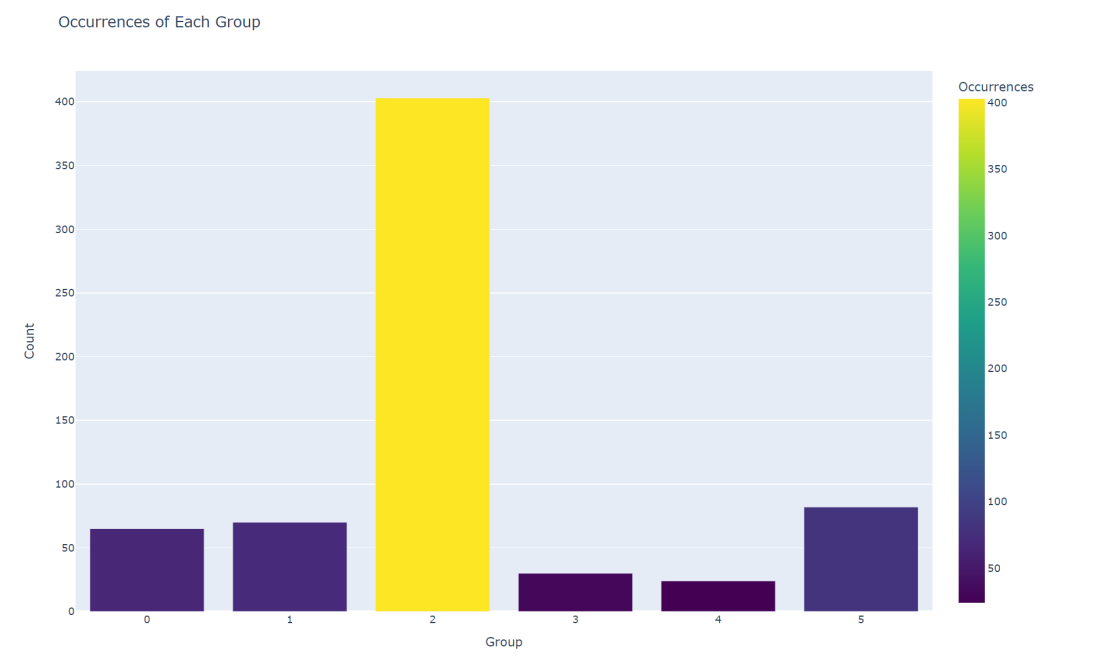
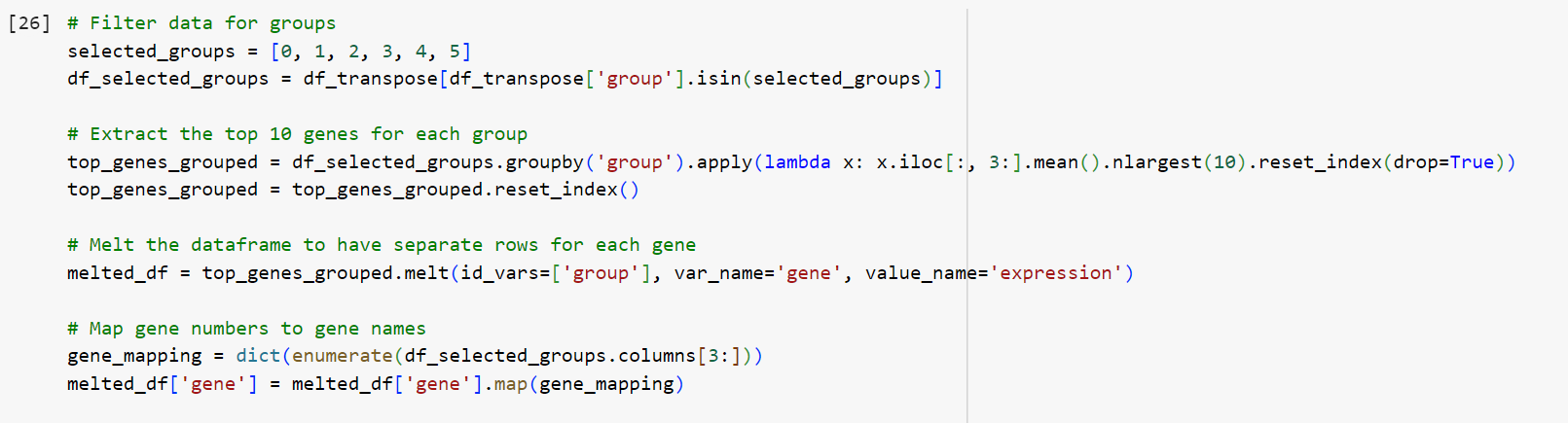
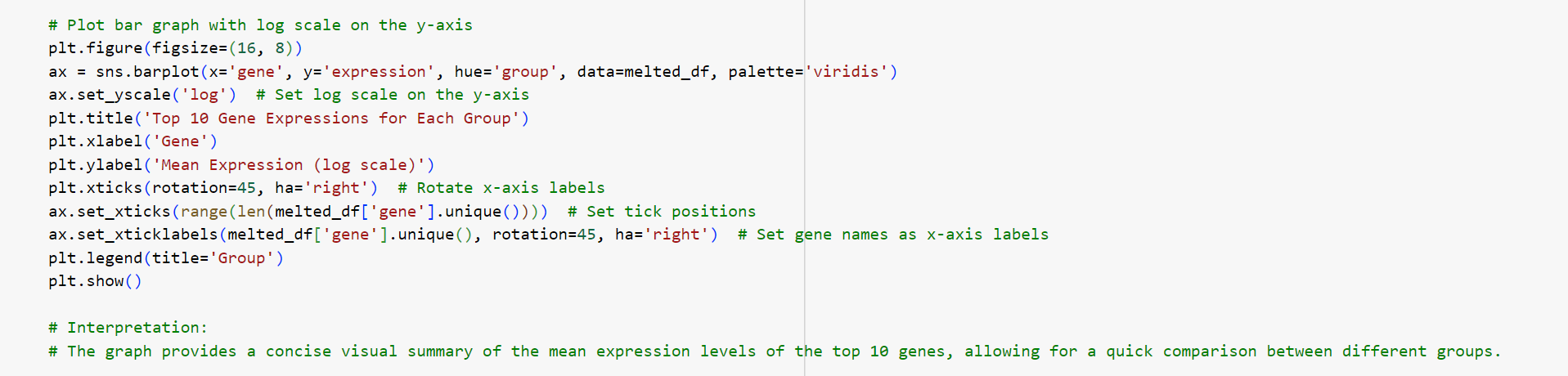
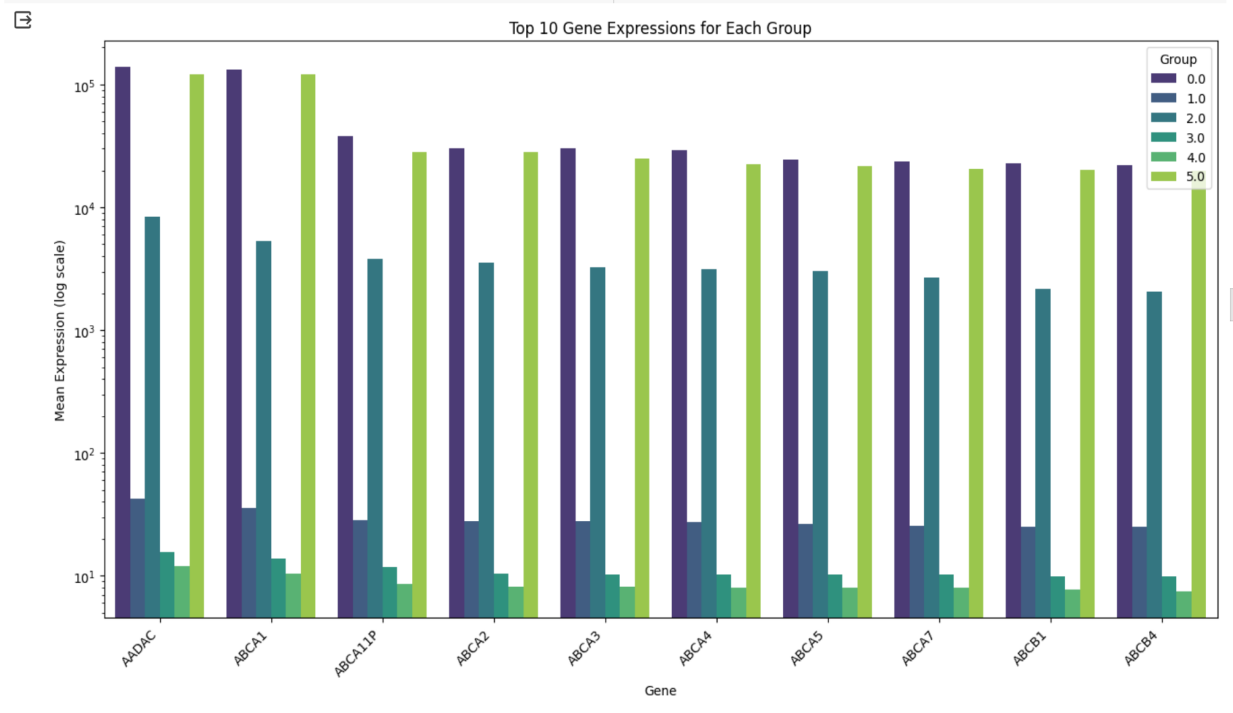
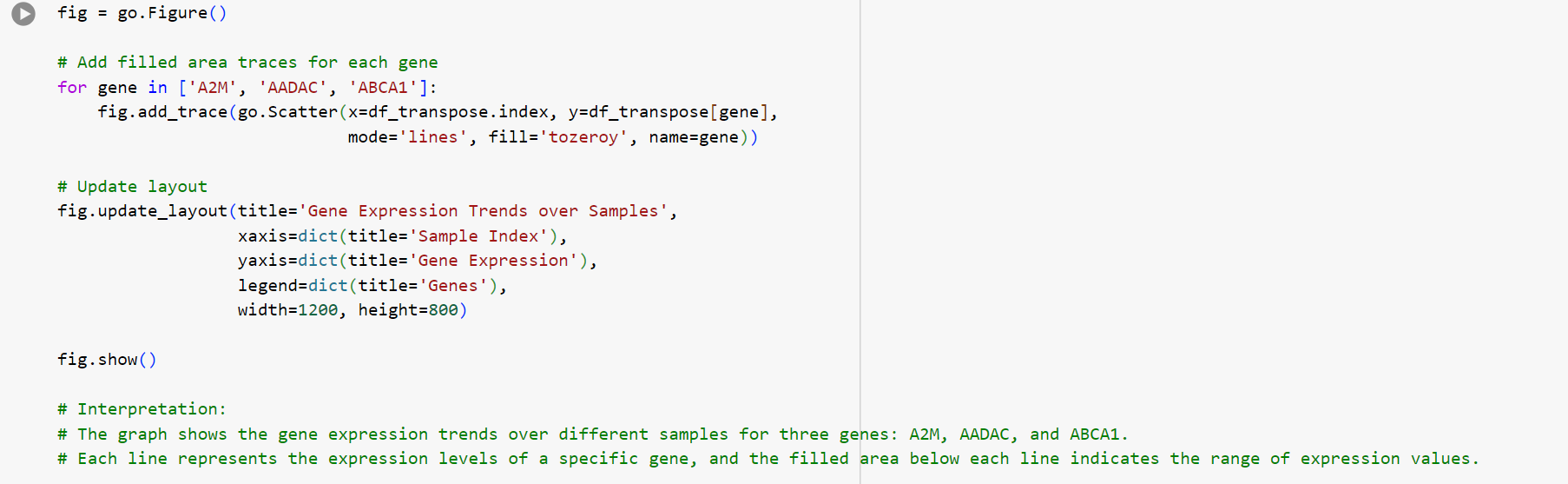
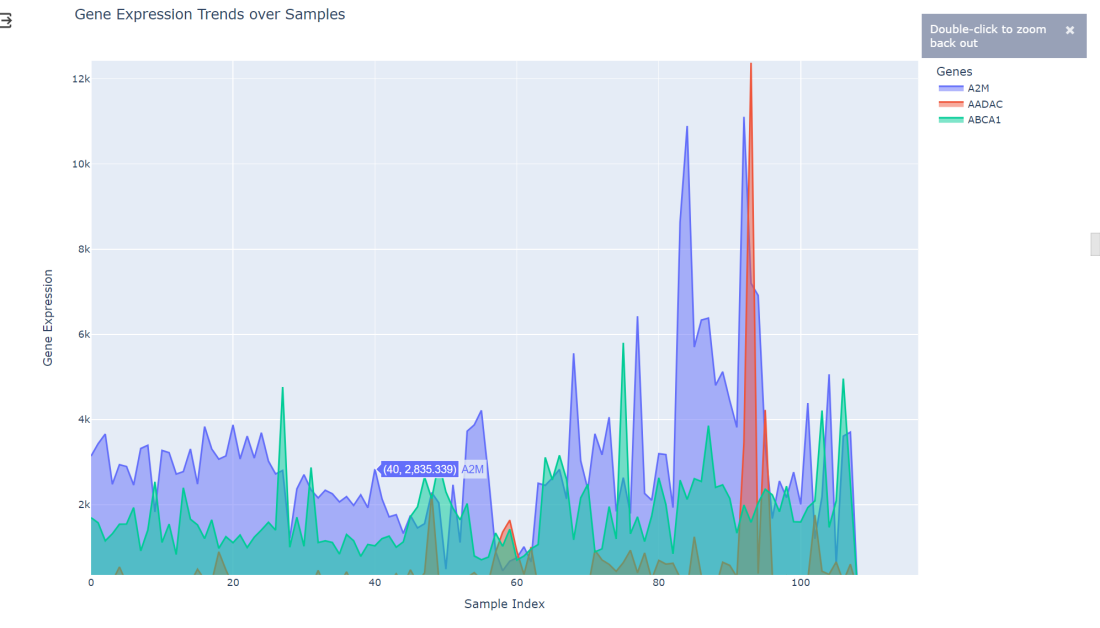
## ***STEP 3a- [Data Preprocessing](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=GIBY2qNCgT4C)***

* Converted categorical group values to numeric representations for effective machine learning.
  + 
* Ensured numeric data types for relevant columns, excluding the 'gene' column.
  + 
  + 
* Grouped values of the same gene name for a given gene type, creating new DataFrames ('df\_sum1' to 'df\_sum7').
  + 
  + 
* Merged all DataFrames based on common genes to create a consolidated DataFrame ('df\_join') by using inner join between the various dataframes.
  + 
  + 

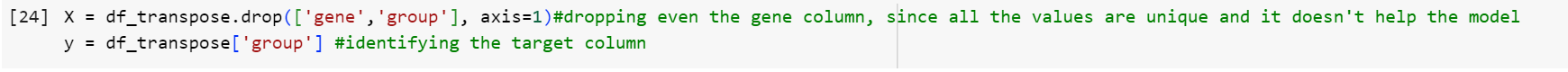
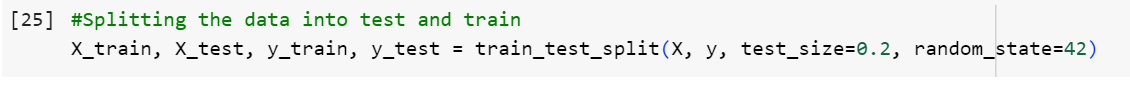
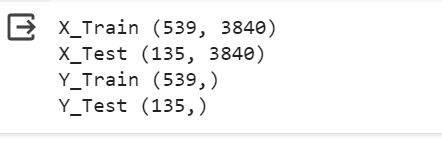
## ***STEP 3b - [Data Transformation](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=nrUbmdUTgrl1)***

* Transposed DataFrames to switch rows and columns, resulting in 'df\_transpose', as we must predict the cancer group which is only possible when the target is in a column rather than row.
* Adjusted column headers, placing gene names as columns and gene types as rows.
* Finalized 'df\_transpose' for downstream analyses, providing a structured and organized dataset.
  + 
  + 

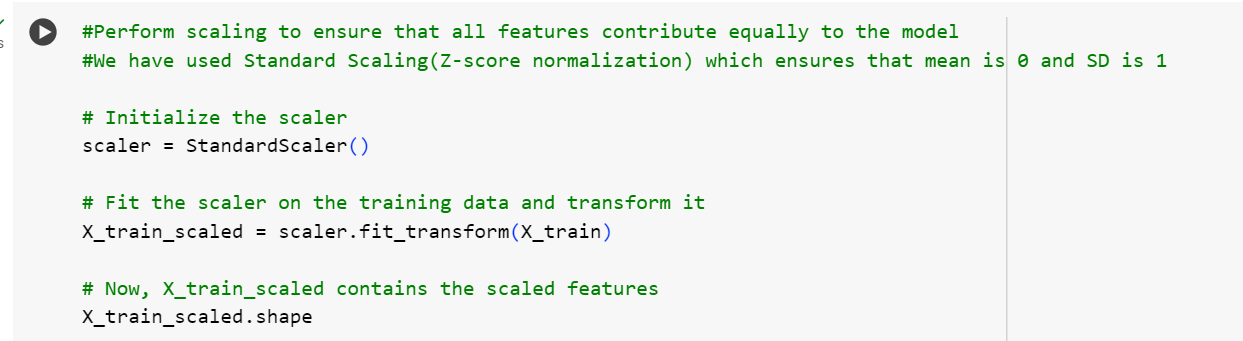
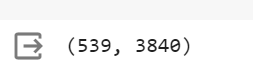
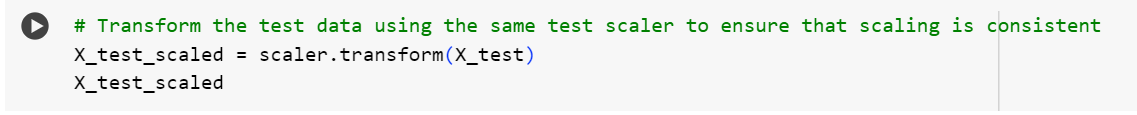
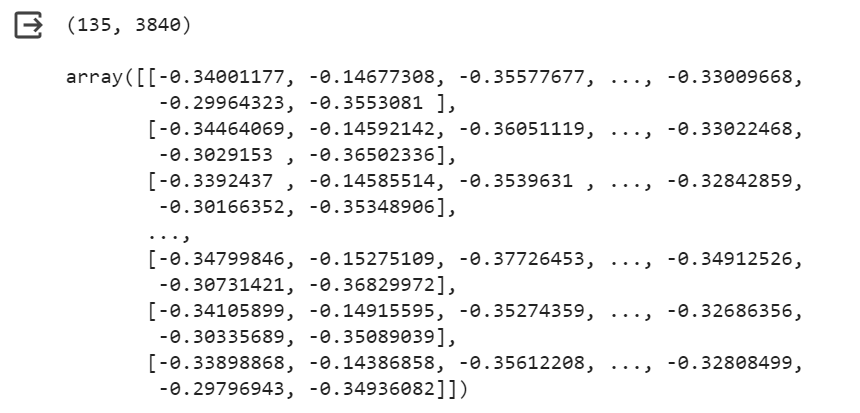
## ***STEP 3c - [Data Visualization](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=IYx1yux4g1_0)***

* **[Scatter Plot of Principal Components](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=DXbCSnFXEHSH&line=5&uniqifier=1)**
  + Applied Principal Component Analysis (PCA) to reduce dimensionality.
    - 
  + Plotted a scatter plot with the first two principal components.
    - 
  + Color-coded points by group for visual interpretation.
* **[Box Plot for a Specific Gene](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=XgcQTvSUIF3R&line=9&uniqifier=1)**
  + Generated a box plot for gene 'A2M' across distinct groups.
    - 
  + Interpreted the plot, observing differences in the median or spread of gene expression values among groups.
  + Identified outliers, if any, represented as individual points outside the whiskers.
    - 
* **[Bar Graph of Group Occurrences](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=ZLRTmT7kPP2j&line=6&uniqifier=1)**
  + Created a bar graph depicting the occurrences of each group.
    - 
  + Quickly assessed the distribution of groups in the dataset.
  + Enabled easy comparison of the occurrence of each group using varying colors.
    - 
* **[Top 10 Gene Expressions for Selected Groups](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=wKRPeIZ1LXMW&line=19&uniqifier=1)**
  + Filtered data for all the groups
  + 
  + Extracted the top 10 genes for each group and visualized mean expression using a bar graph with a log scale on the y-axis.
  + 
  + 
  + Facilitated a quick comparison of gene expressions among separate groups.
  + 
* **[Gene Expression Trends Over Samples](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=rpD4wBIbhJFD&line=9&uniqifier=1)**
  + Plotted gene expression trends for genes A2M, AADAC, and ABCA1 over different samples.
    - 
  + Utilized a filled area plot to represent the range of expression values for each gene.
  + Provided a visual representation of how gene expression varies across samples.
    - 

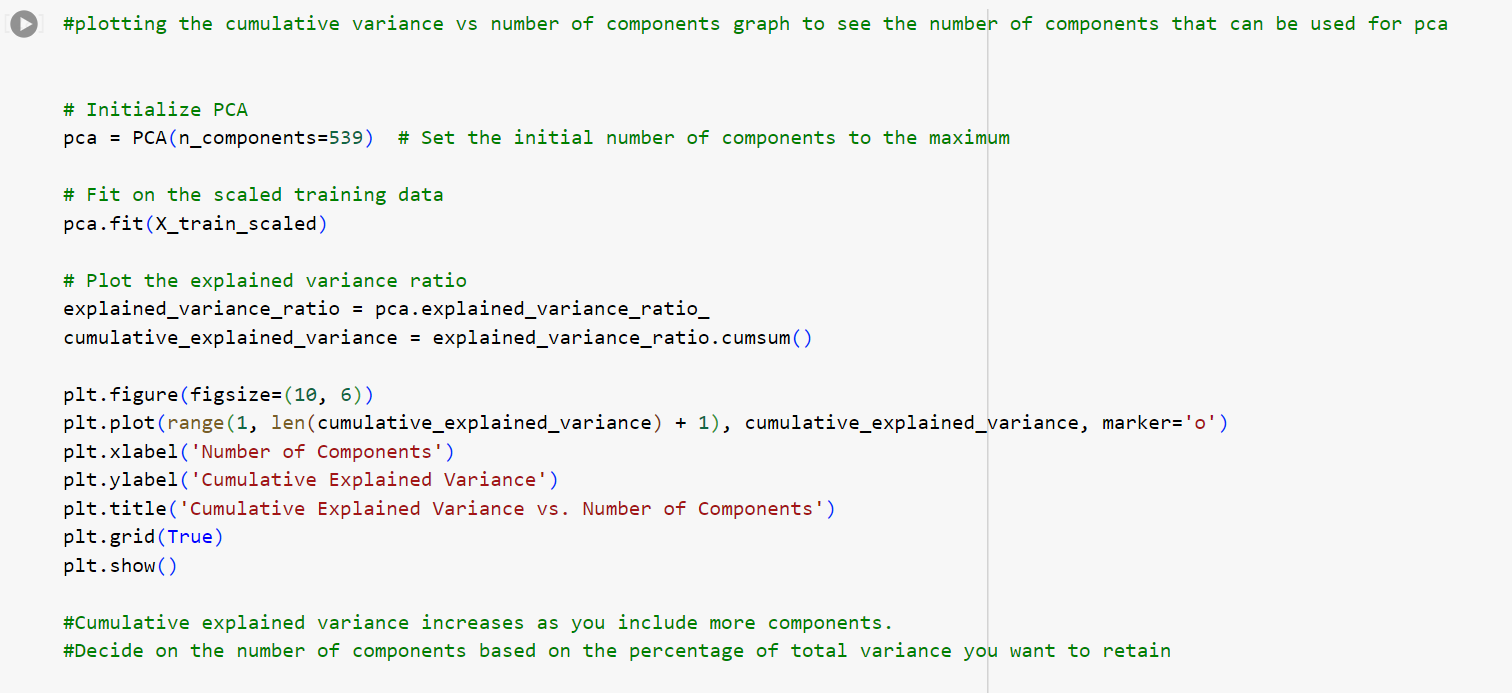
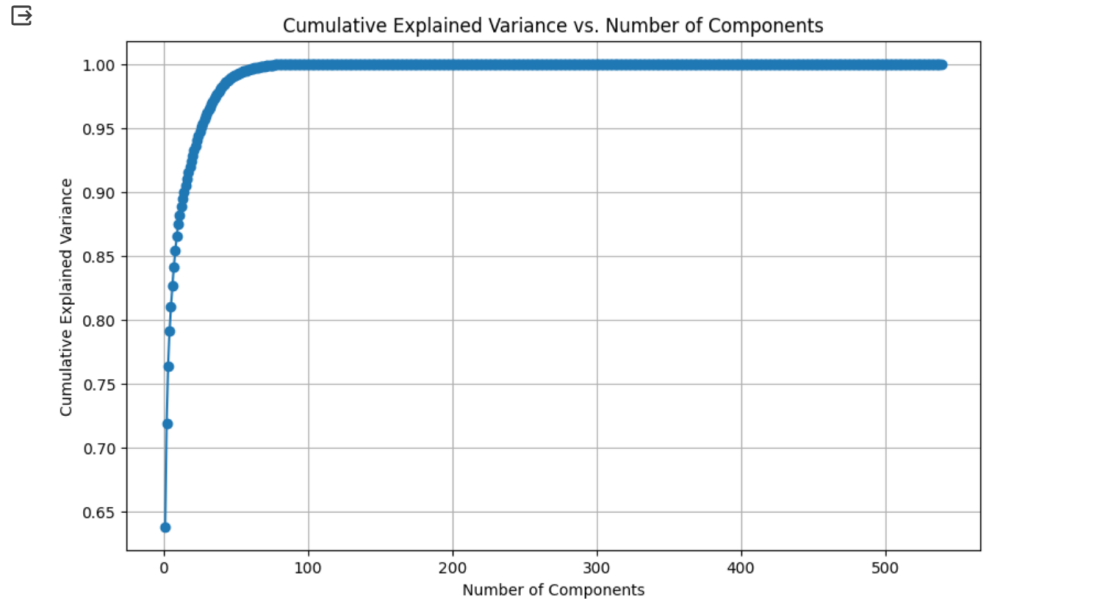
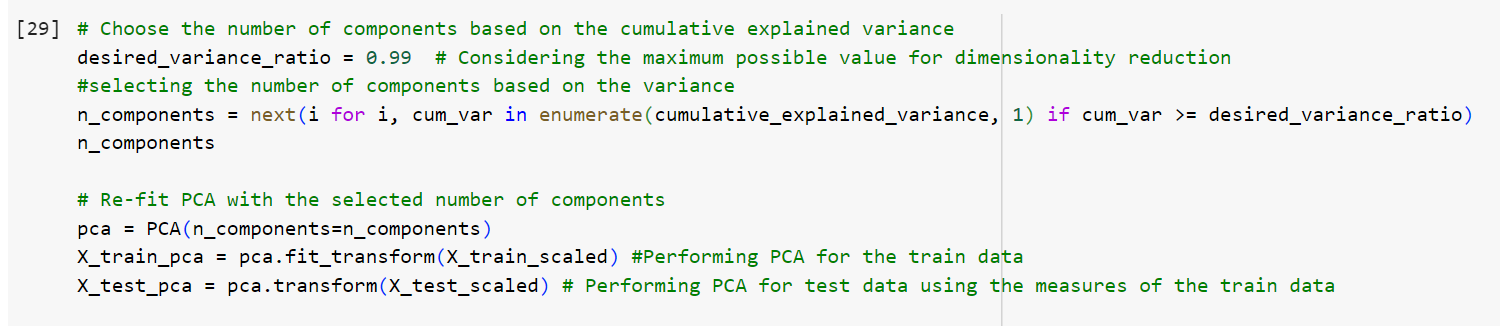
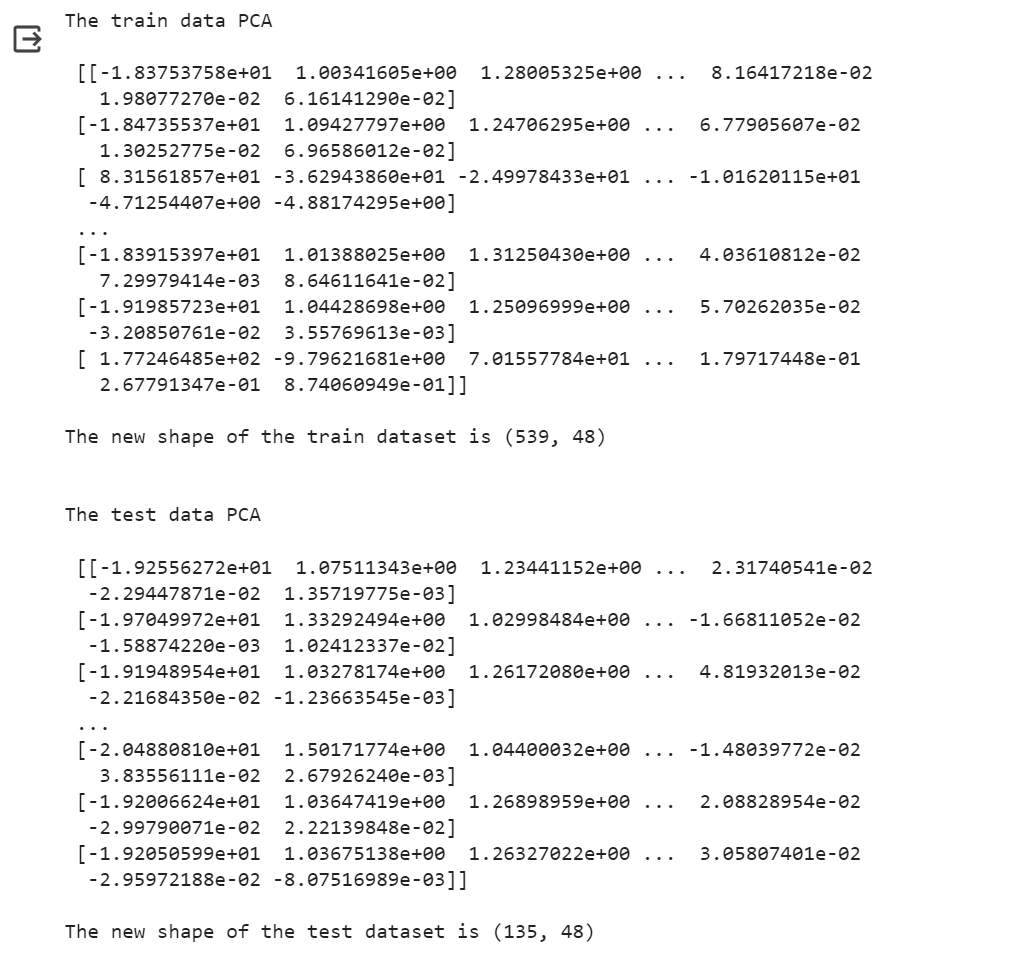
## ***STEP 3d - [Data Splitting](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=fE2PqdDKhHQ7)***

* Dropped 'gene' column as it has unique values, and 'group' column was identified as the target.
  + 
* Split the data into training and testing sets using a test size of 20% and a random seed for reproducibility.
  + 
  + 

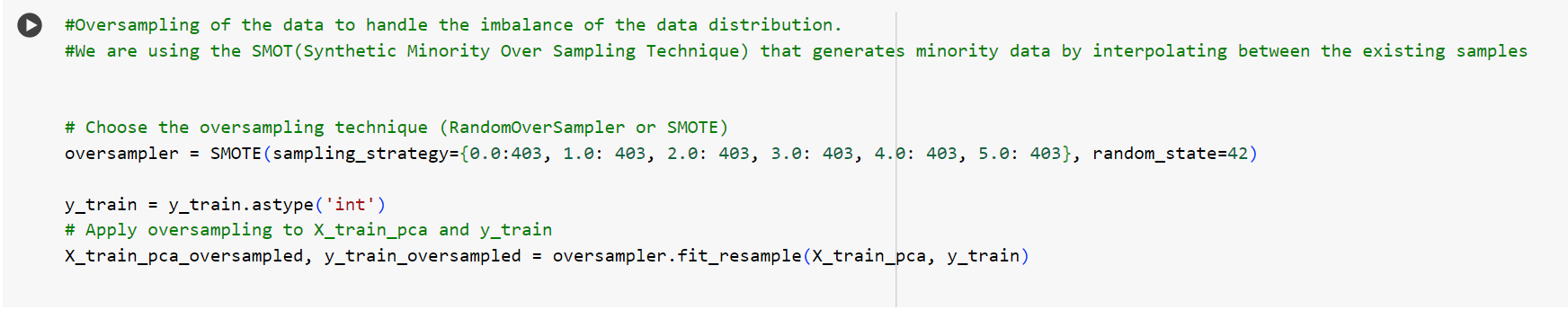
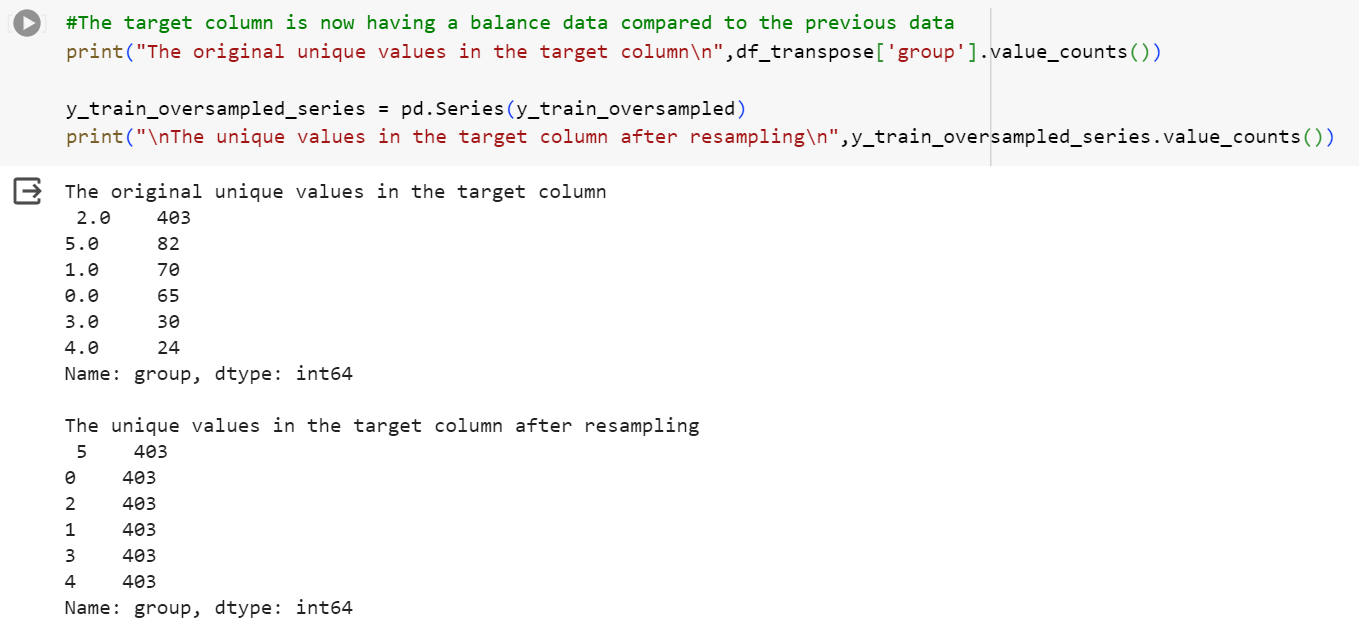
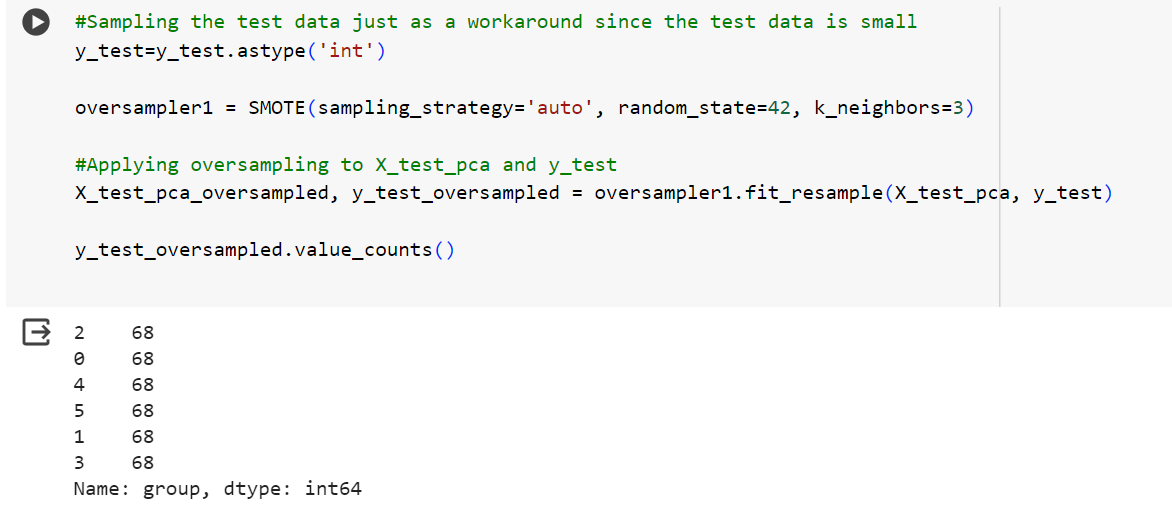
## ***STEP 3e - [Feature Scaling](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=QuPsdk33hXAi)***

* Applied Standard Scaling (Z-score normalization) to ensure all features contribute equally.
  + 
  + 
* Scaled the training data and transformed the test data using the same scaler for consistency.
  + 
  + 

## ***STEP 3e - [Principle Component Analysis (PCA)](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=dT6qsB_uhl6H)***

* Explored the cumulative explained variance vs. the number of components to decide on the optimal number of dimensions required for the dimensionality reduction.
  + 
* Selected the number of components to retain 99% of the total variance.
* 
* Refitted PCA with the chosen number of components.
* Transformed the training and test data using the PCA.
  + 
  + 

## ***STEP 3f - [Oversampling for Imbalanced Data](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=xIc0K0HKh0tq)***

* Addressed imbalanced data distribution using Synthetic Minority Over Sampling Technique (SMOTE).
* Applied SMOTE to the training data to balance the class distribution.
  + 
* Checked and confirmed the balanced distribution in the target column after oversampling.
  + 
* Applied SMOTE to the test data as a workaround for its small size, maintaining its balance.
  + 

## ***STEP 4 – Model Prediction***

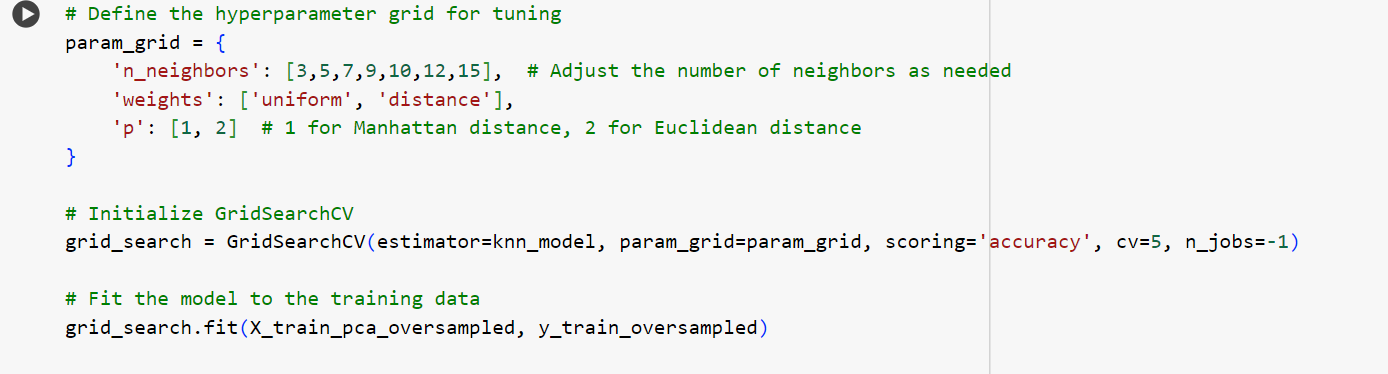
## Step 4a - [K-Nearest Neighbors (KNN)](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=O5A2HmFzAM5O&line=1&uniqifier=1)

K-Nearest Neighbors (KNN) is a non-parametric, supervised machine learning algorithm used for classification and regression tasks. It operates on the principle that similar data points in a feature space tend to share common characteristics. In the context of classification, when predicting the class of a new data point, KNN identifies the K nearest neighbors to that point based on a chosen distance metric. The most prevalent class among these neighbors is then assigned to the new data point. KNN's simplicity and intuitive concept make it a valuable tool for various applications, particularly when the underlying data distribution is not well-defined or exhibits complex patterns.

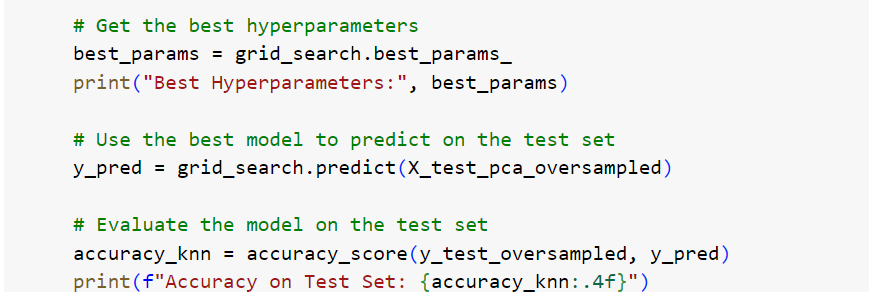
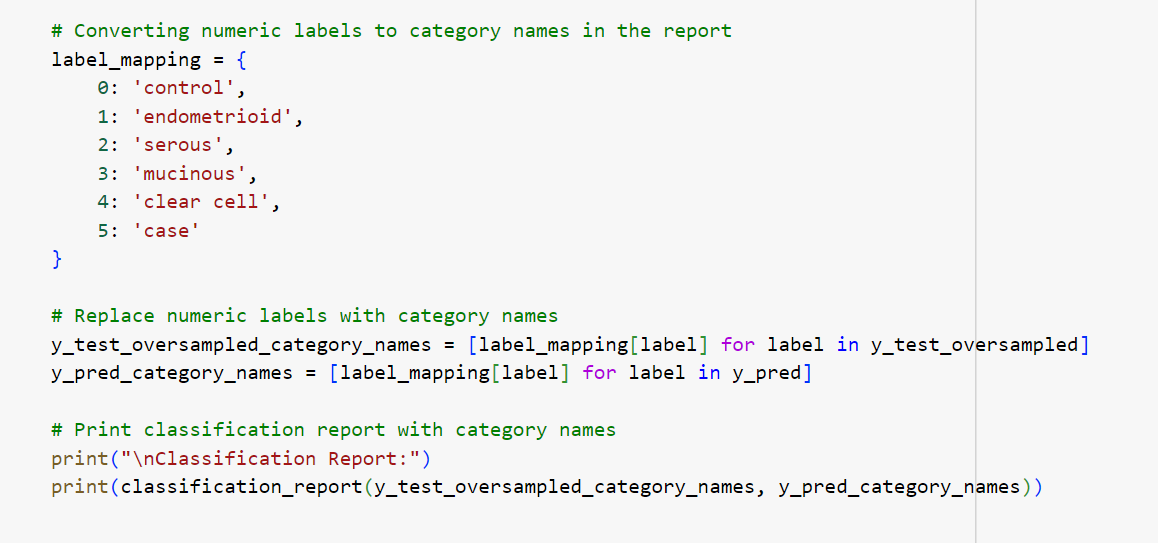
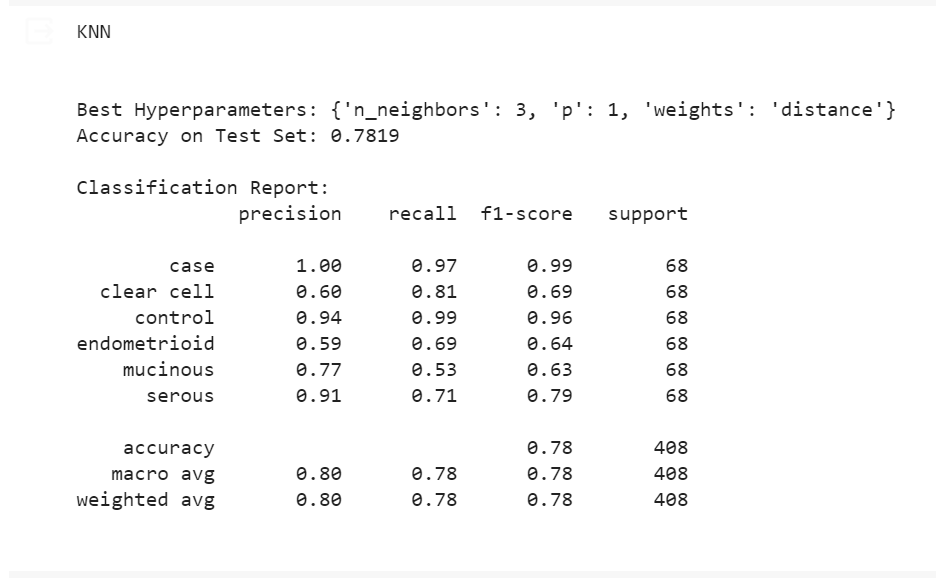
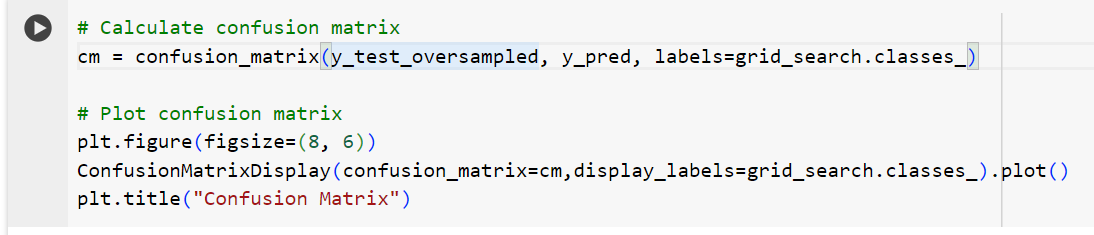
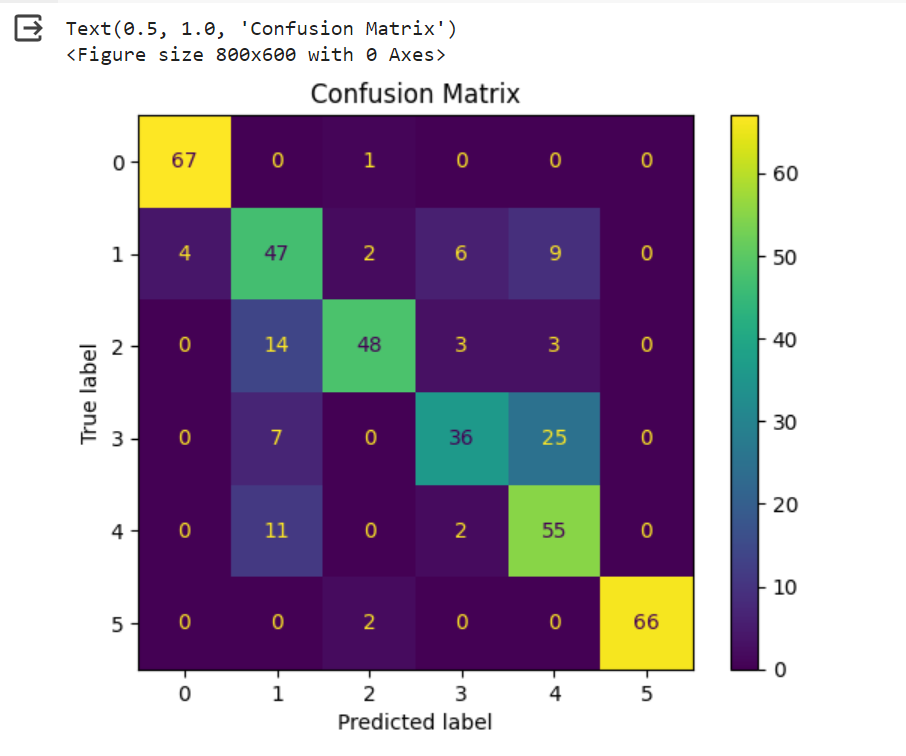
**Model Training**

* Initialized and trained a KNN model on the oversampled training data.
* Configured the model with a specified number of neighbors (n\_neighbors) during initialization.
* Utilized the trained model to predict on the oversampled test set.
* Evaluated model accuracy on the test set.

**Hyperparameter Tuning**

* Defined a hyperparameter grid for tuning, including the number of neighbors, weight function, and distance metric (p).
* Utilized GridSearchCV to find the best combination of hyperparameters.
  + 
* Fit the model to the training data using the optimized hyperparameters.
* Retrieved and printed the best hyperparameters.

**Model Evaluation with Tuned Hyperparameters**

* Used the tuned KNN model to predict on the oversampled test set.
* Re-evaluated the model accuracy on the test set with the optimized hyperparameters.
  + 
* Printed a classification report for a detailed performance summary.
  + 
* The classification report includes precision, recall, F1-score, and support for each class.
  + 
* Used the test data to display a confusion matrix based on the model prediction
  + 
  + 

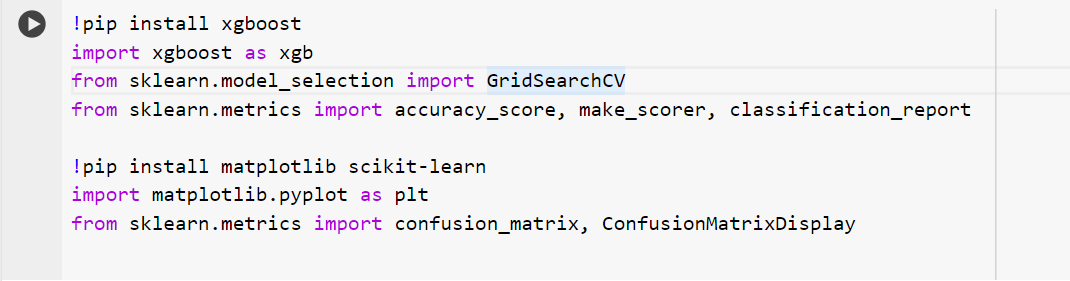
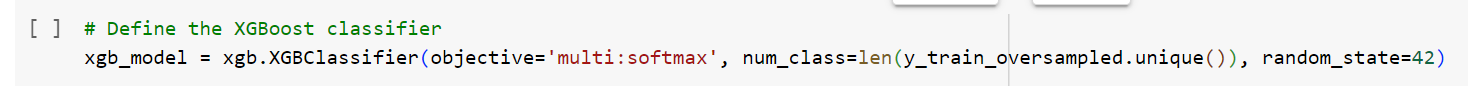
Note:

* KNN is sensitive to the choice of hyperparameters and tuning them can significantly impact model performance.
* The number of neighbors (n\_neighbors) determines the number of data points considered for classification.
* Weight function ('uniform' or 'distance') influences the contribution of neighbors to the classification.
* The distance metric (p) can be set to 1 for Manhattan distance and 2 for Euclidean distance.

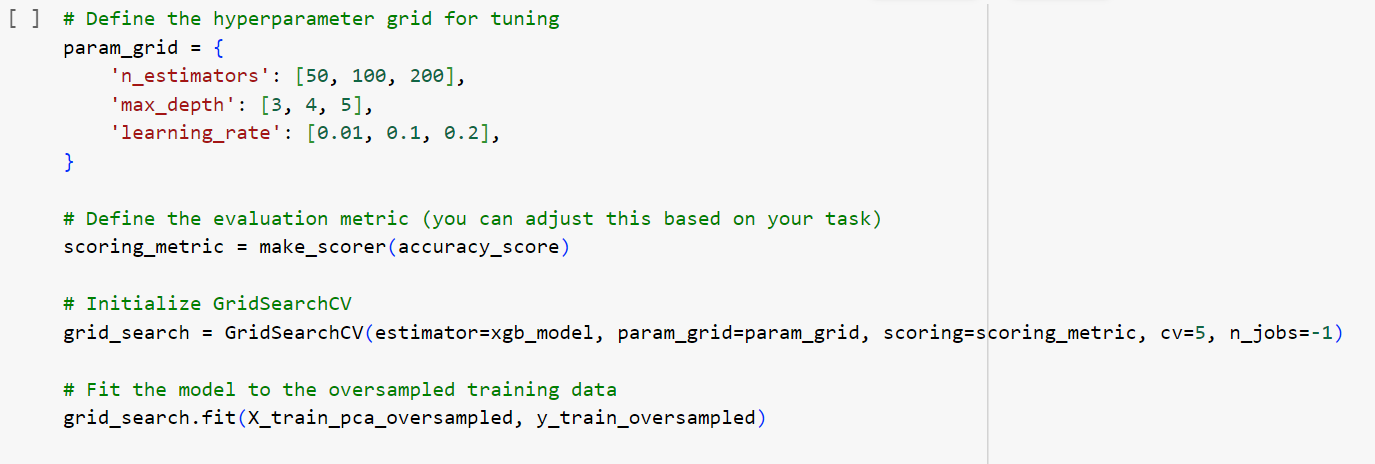
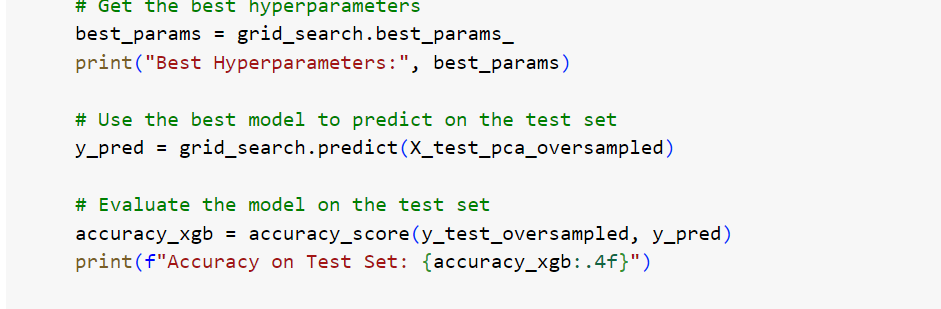
## Step 4b - [XGBoost](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=4j6F10y5LKVq&line=1&uniqifier=1)

XGBoost, or eXtreme Gradient Boosting, is an advanced and scalable machine learning algorithm that belongs to the ensemble learning family. Specifically designed for tree-based models, XGBoost combines the strengths of gradient boosting with regularization techniques to enhance predictive performance. It sequentially builds a multitude of decision trees, where each subsequent tree corrects the errors of the previous ones. XGBoost incorporates regularization terms to control model complexity, preventing overfitting and improving generalization. Known for its efficiency, speed, and ability to handle large datasets, XGBoost has become a popular choice in various machine learning competitions and real-world applications. Its flexibility allows for both classification and regression tasks, making it a versatile and powerful tool in the machine learning toolkit.

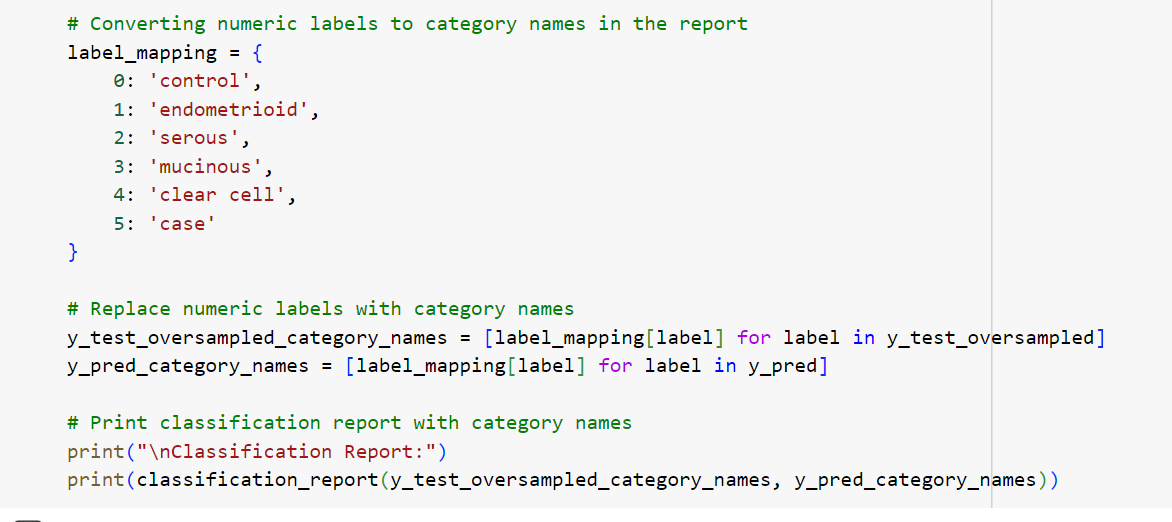
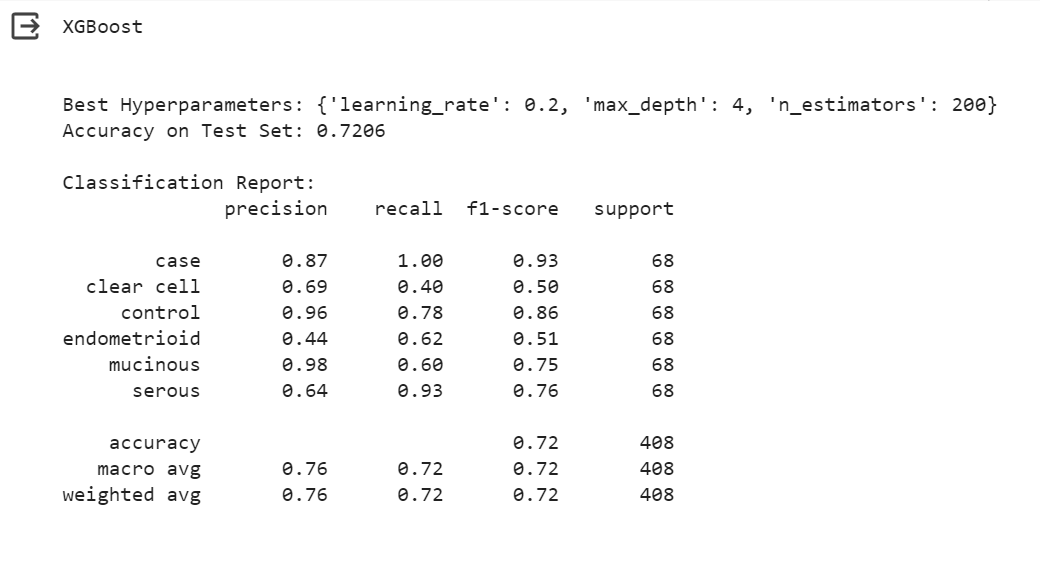
**Model Training**

* Installed and imported necessary libraries, including XGBoost, for model training and evaluation.
  + 
* Defined an XGBoost classifier with parameters suitable for multiclass classification.
  + 
* Specified hyperparameters for tuning, such as the number of estimators, maximum depth, and learning rate.

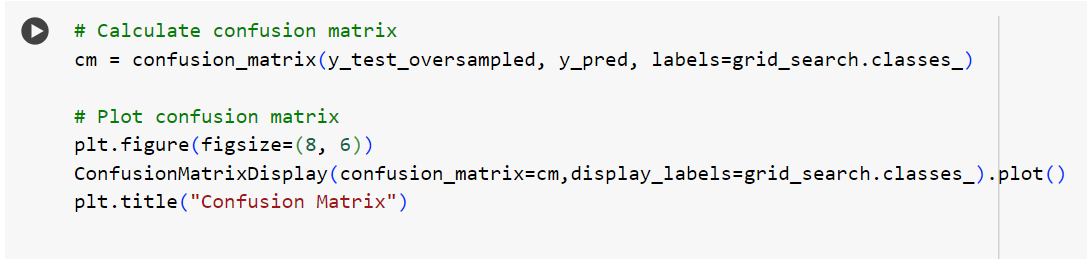
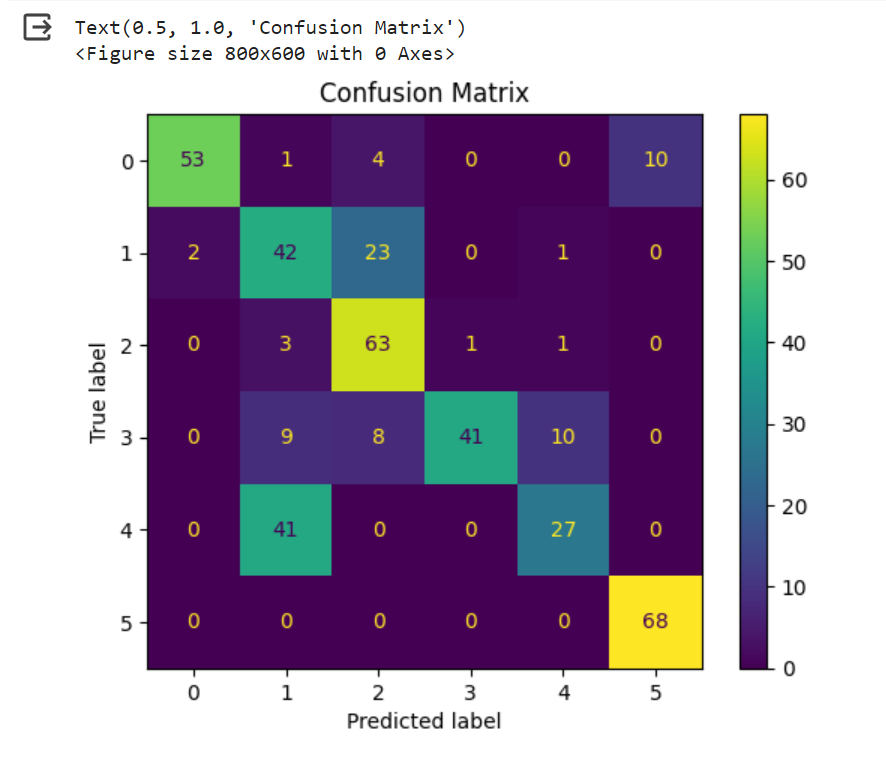
**GridSearchCV for Hyperparameter Tuning**

* Set up a hyperparameter grid for tuning.
* Used GridSearchCV to perform a search across the hyperparameter space.
  + 
* Employed accuracy as the evaluation metric for tuning.
* Obtained the best hyperparameters for the XGBoost model.
  + 

**Model Evaluation**

* Fitted the XGBoost model to the oversampled training data.
* Predicted on the oversampled test set and evaluated accuracy.
* Utilized a classification report for a detailed performance summary.
  + 
  + 
* Conducted cross-validation to assess model performance across different subsets of the test data.

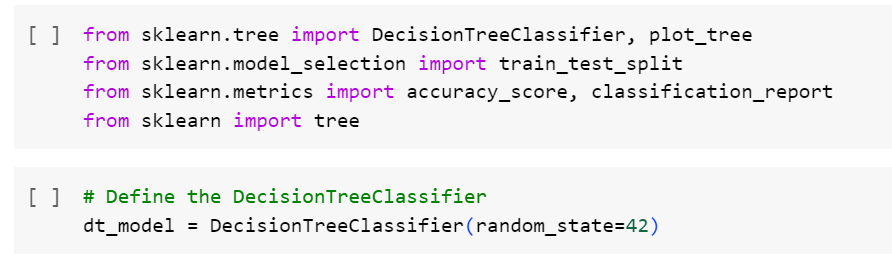
**Confusion Matrix Visualization**

* Calculated the confusion matrix to understand model classification performance.
* Plotted the confusion matrix using matplotlib and ConfusionMatrixDisplay.
  + 
  + 

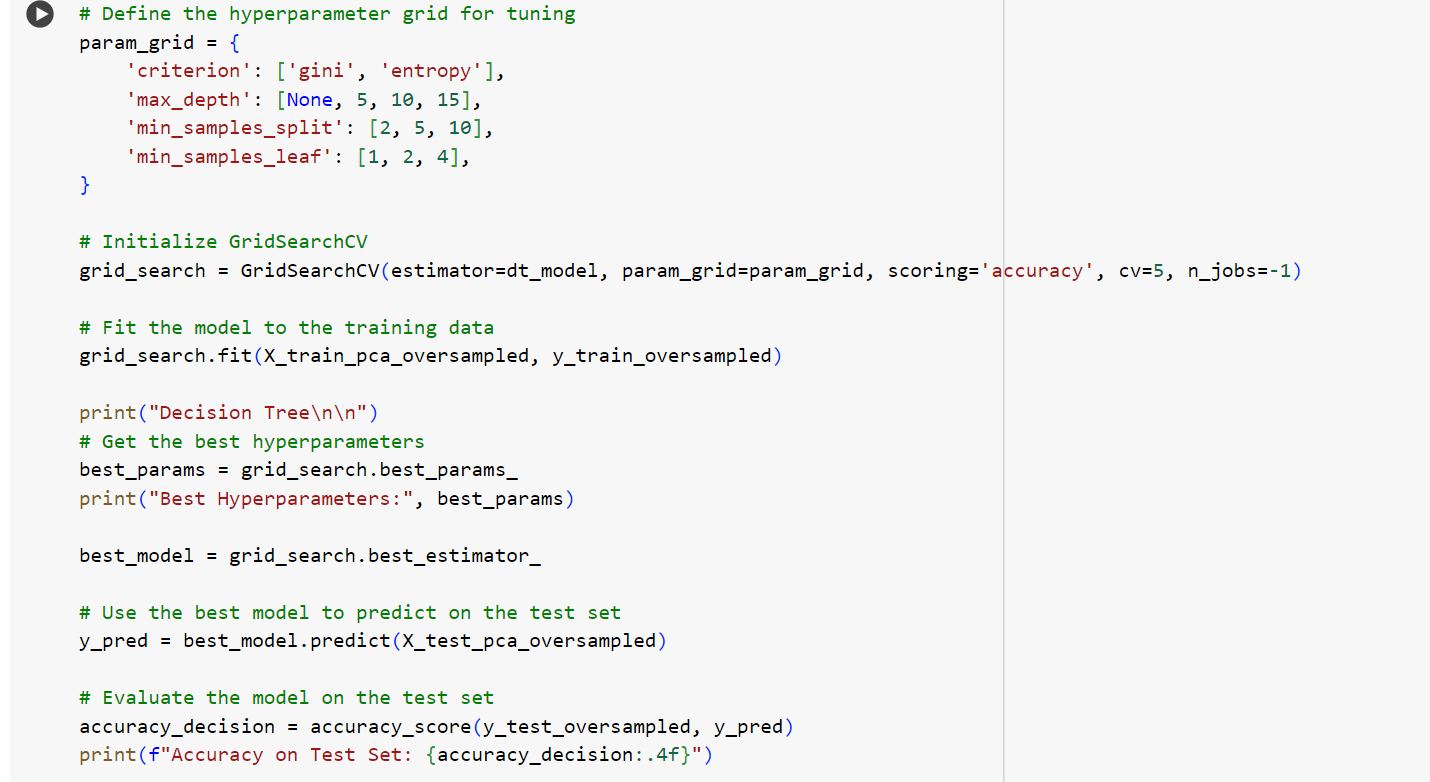
## Step 4c - [Decision Tree](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=fM1weRJFtCJq&line=1&uniqifier=1)

A Decision Tree is a fundamental machine learning algorithm used for both classification and regression tasks. It models decisions as a tree-like structure, where each internal node represents a decision based on a feature, each branch corresponds to the outcome of the decision, and each leaf node represents the final predicted label or value. The algorithm recursively splits the data based on features, making decisions that optimize a selected criterion, such as Gini impurity or information gain. Decision Trees are transparent and interpretable, making them valuable for understanding the decision-making process. However, they are prone to overfitting, and techniques like pruning are employed to mitigate this. Ensemble methods, such as Random Forests, often utilize multiple decision trees to enhance predictive accuracy and robustness.

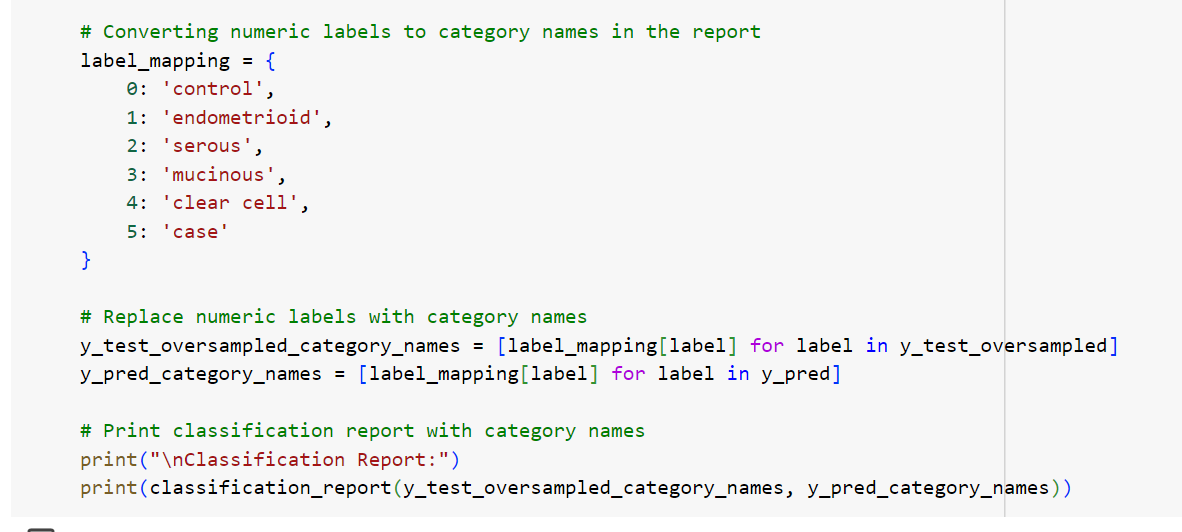
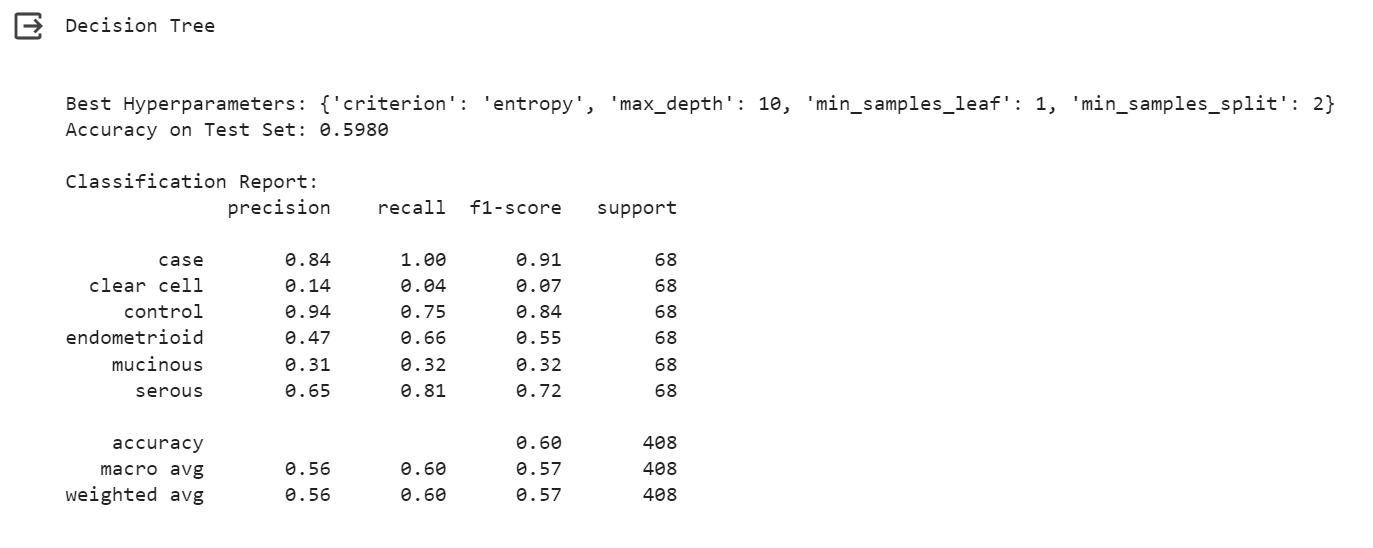
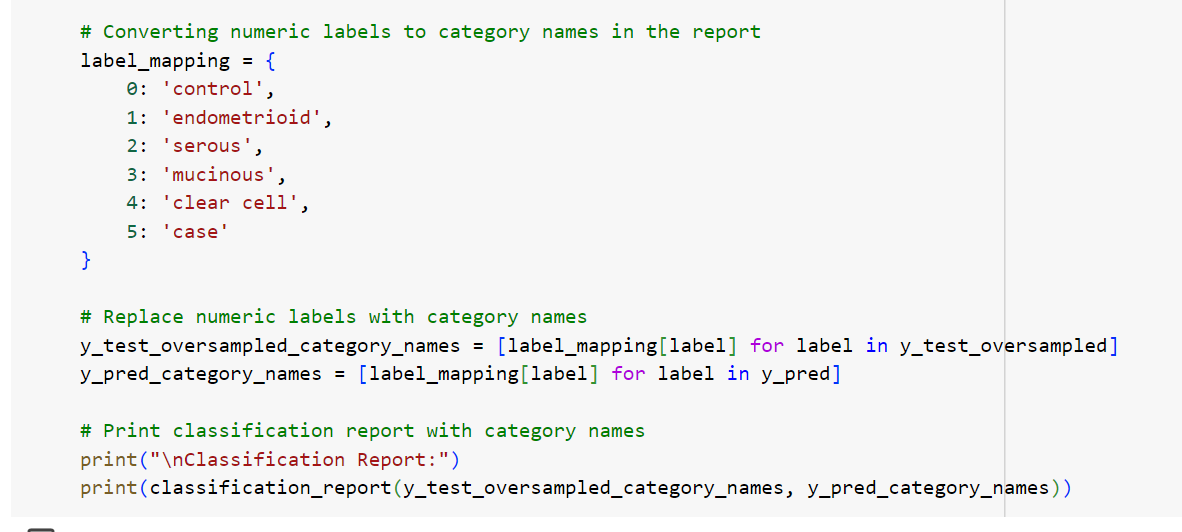
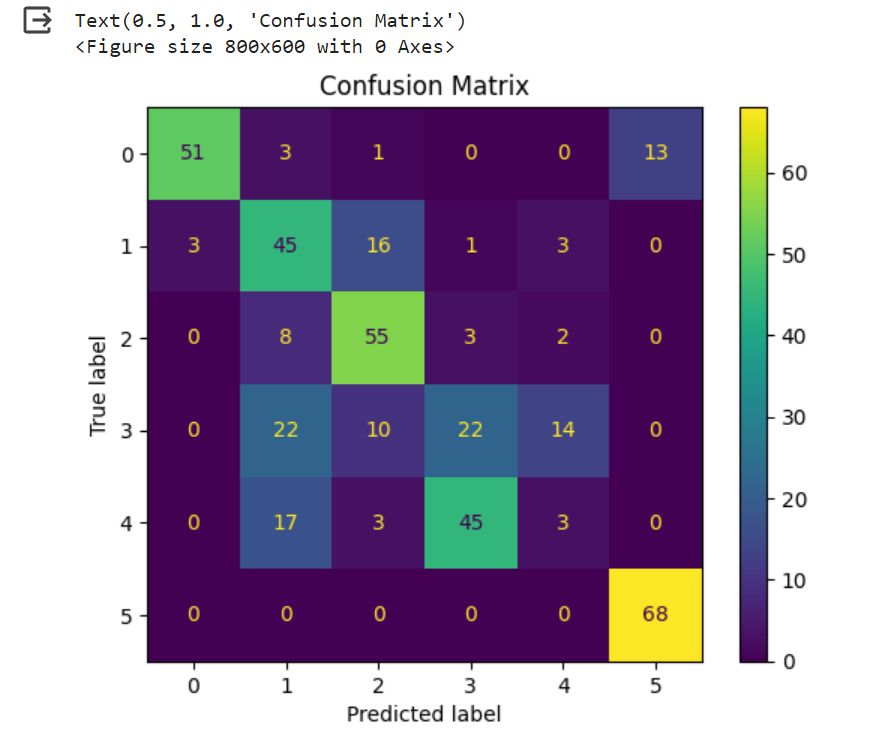
**Model Training**

* Initialized and trained a Decision Tree Classifier on the oversampled training data.
  + 
* Utilized the trained model to predict on the oversampled test set.
* Evaluated model accuracy on the test set.

**Hyperparameter Tuning**

* Defined a hyperparameter grid for tuning, including criteria, maximum depth, minimum samples split, and minimum samples leaf.
* Used GridSearchCV to perform a search across the hyperparameter space.
  + 
* Obtained the best hyperparameters for the Decision Tree model.

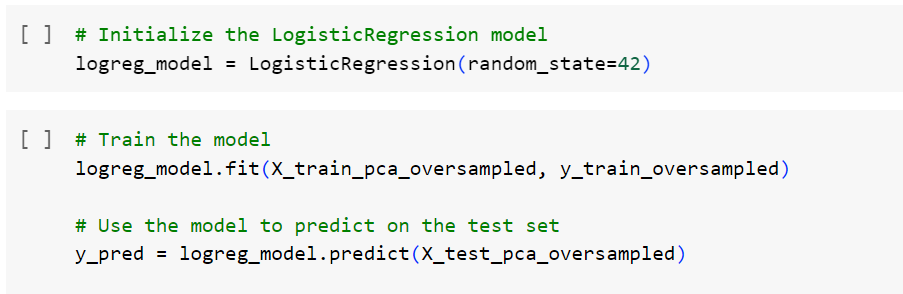
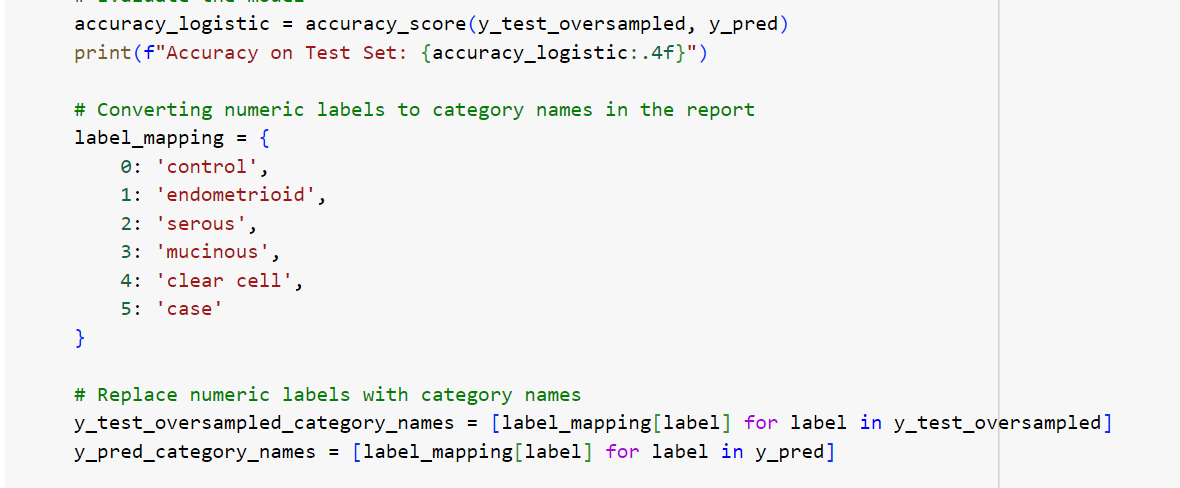
**Model Evaluation**

* Fitted the Decision Tree model with the best hyperparameters to the oversampled training data.
* Predicted on the oversampled test set using the tuned model.
* Evaluated model accuracy on the test set.
* Printed a classification report for a detailed performance summary.
  + 
  + 
* Calculated the confusion matrix to understand model classification performance.
  + 
  + 

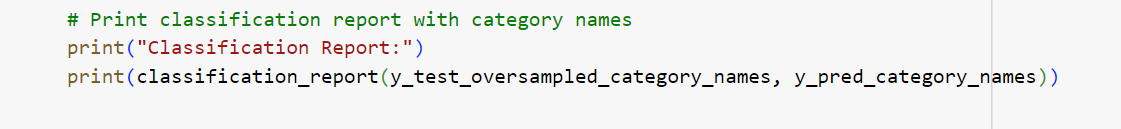
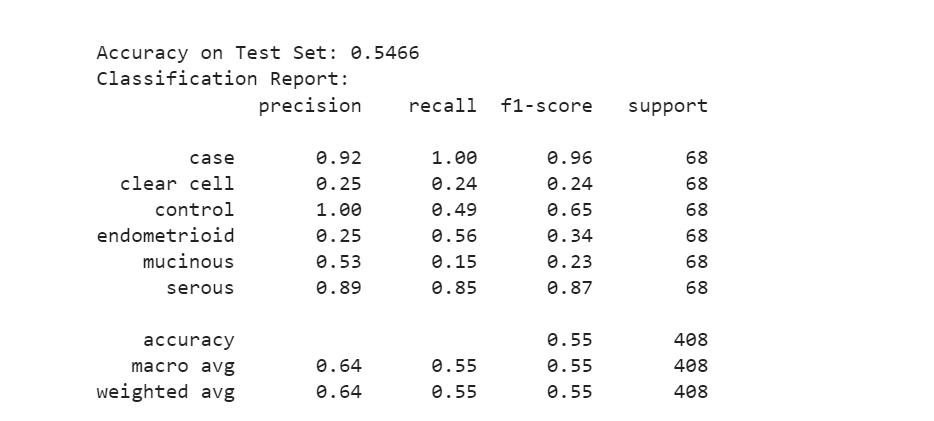
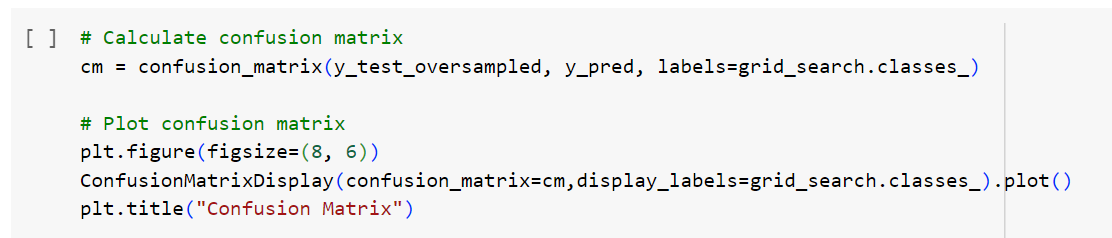
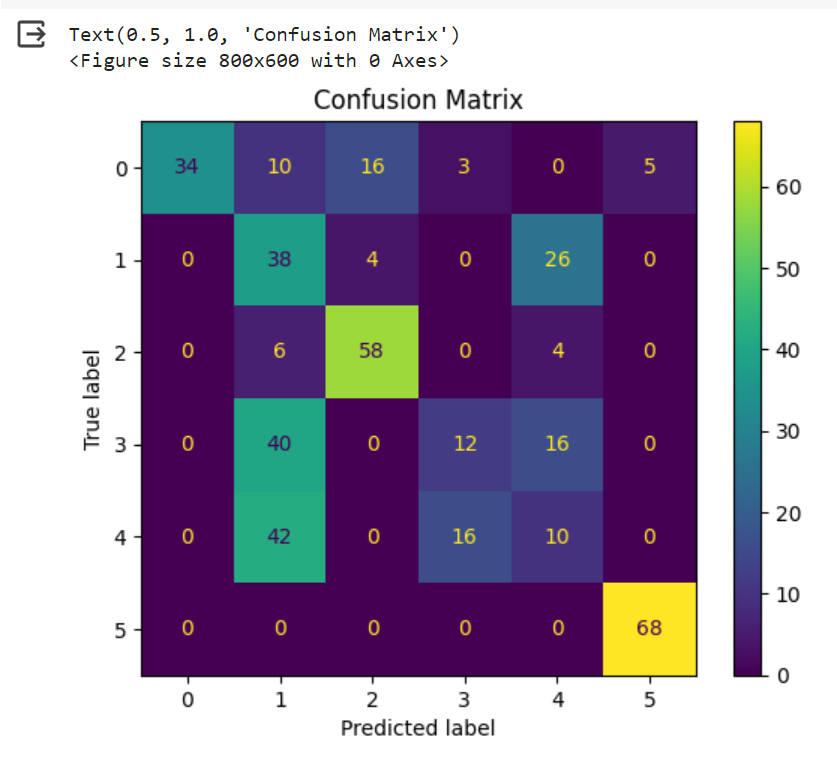
## Step 4d - [Logistic Regression](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=W55CCyw99W8P&line=1&uniqifier=1)

Logistic Regression is a classification algorithm commonly used for binary classification problems. It predicts the probability that an instance belongs to a particular class using the logistic function. The algorithm models the relationship between the features and the binary outcome by estimating coefficients for each feature. These coefficients are then applied to a logistic function, transforming the output into a probability score between 0 and 1. A decision boundary is set, and instances with probabilities above a certain threshold are classified into one category, while those below are classified into the other. Logistic Regression is widely employed due to its simplicity, efficiency, and interpretability. It's sensitive to linear relationships between features and the target variable, making it particularly suitable for well-behaved datasets.

**Model Training**

* Initialized and trained a Logistic Regression model on the oversampled training data.
  + 
* Utilized the trained model to predict on the oversampled test set.
  + 
* Evaluated model accuracy on the test set.

**Model Evaluation**

* Printed a classification report for a detailed performance summary.
  + 
* The classification report includes precision, recall, F1-score, and support for each class.
  + 
* Precision represents the ratio of true positive predictions to the total positive predictions.
* Recall indicates the ratio of true positive predictions to the total actual positives.
* F1-score is the harmonic mean of precision and recall, providing a balanced measure.
* Support represents the number of actual occurrences of each class in the specified dataset.
* Calculated the confusion matrix to understand model classification performance.
  + 
  + 

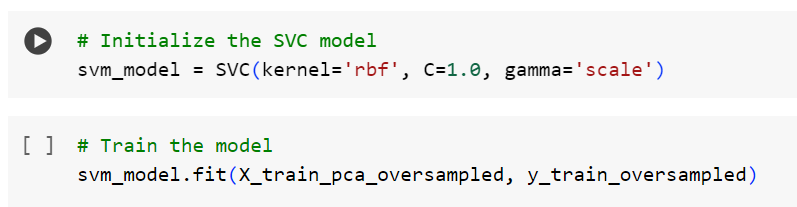
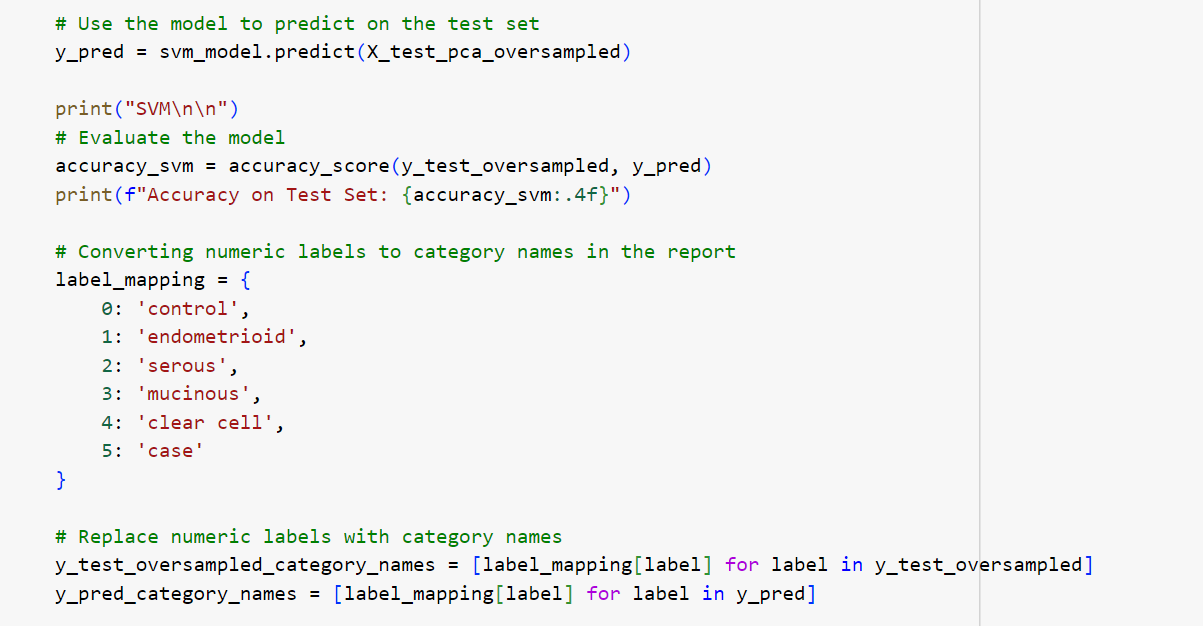
Note:

* Logistic Regression is particularly useful for binary classification problems, but it can be extended to multi-class problems using the 'multi\_class' parameter.
* The 'random\_state' parameter ensures reproducibility by fixing the seed for random number generation.

## [Support Vector Machine (SVM)](https://colab.research.google.com/drive/11SXILh8eREX4YOSE_Z9uwYk7X5sWHhW0" \l "scrollTo=NvzQmansGpJi&line=1&uniqifier=1)

Support Vector Machines are powerful supervised learning algorithms used for both classification and regression tasks. SVM aims to find the optimal hyperplane that best separates data points of different classes in a high-dimensional space. The "support vectors" are the instances located closest to the decision boundary, influencing its placement. SVM can handle non-linear relationships by mapping the input features into a higher-dimensional space through a kernel function. This allows SVM to identify more complex decision boundaries. SVMs are effective in high-dimensional spaces and are particularly useful when the classes are not linearly separable. They excel in scenarios with clear margins between classes and are robust against overfitting, making them versatile for various applications.

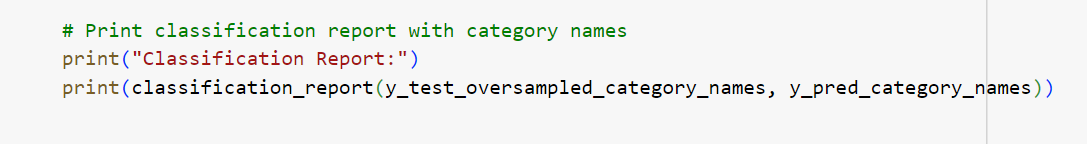
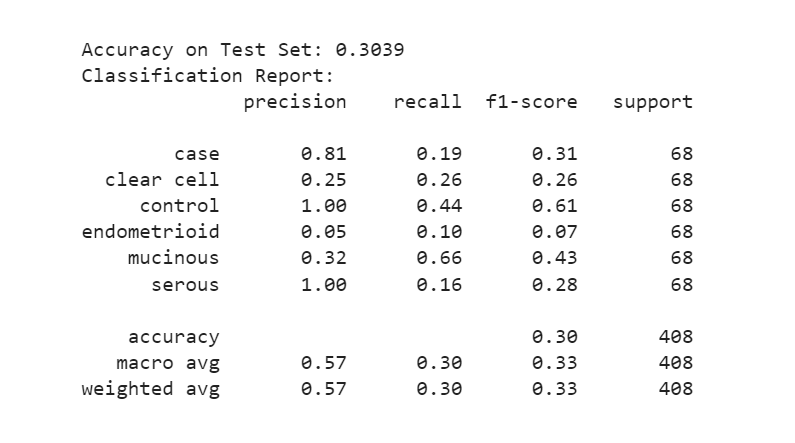
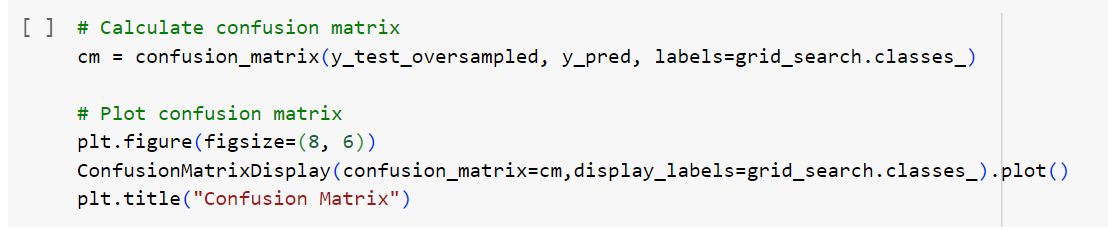
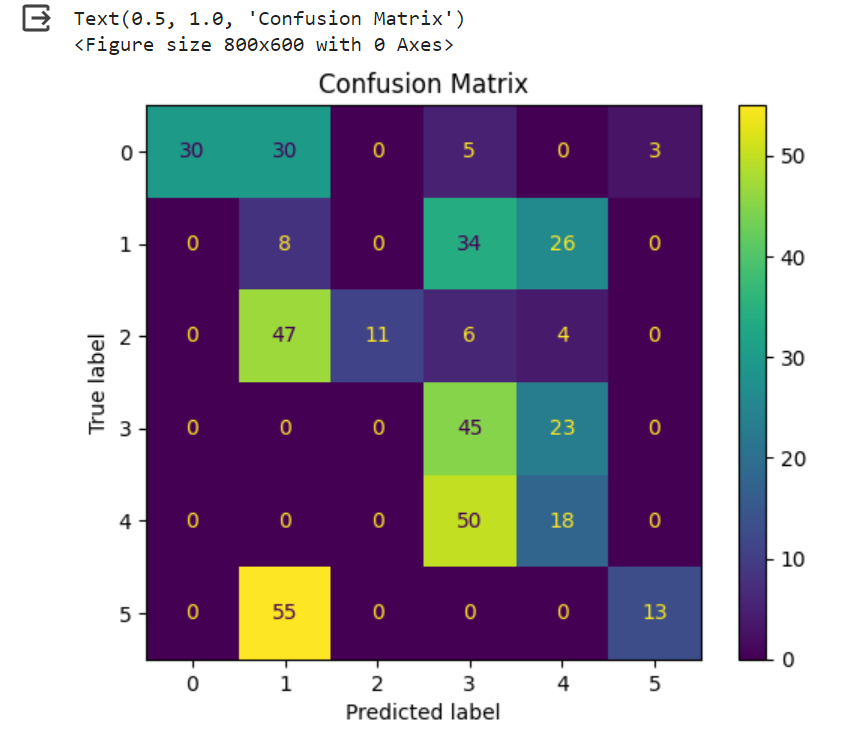
**Model Training**

* Initialized and trained an SVM model on oversampled training data.
  + 
* Configured the model with a radial basis function (RBF) kernel, regularization parameter (C), and gamma value during initialization.
* Utilized the trained SVM model to predict on the oversampled test set.
* Evaluated model accuracy on the test set.
  + 

**Hyperparameter Configuration:**

* Kernel: RBF (Radial Basis Function) is a common choice, but other kernels like linear or polynomial can be explored.
* Regularization Parameter (C): Controls the trade-off between achieving a smooth decision boundary and classifying training points correctly. A higher C value emphasizes accurate classification of training points.
* Gamma: Defines how far the influence of a single training example reaches. A low gamma indicates a 'far' reach, and a high gamma indicates a 'close' reach.

**Model Evaluation:**

* Printed the accuracy of the SVM model on the oversampled test set.
* Generated a classification report for a detailed performance summary.
  + 
* The classification report includes precision, recall, F1-score, and support for each class.
  + 
* Calculated the confusion matrix to understand model classification performance.
  + 
  + 

Note:

* SVMs are versatile and can handle both linear and non-linear classification tasks.
* The choice of the kernel and hyperparameters significantly affects SVM performance.
* Hyperparameter tuning can be performed for optimization based on the specific dataset and task.

# **Description of the Dataset**

The dataset being used is about the Ovarian tumors collected from the experiment data. The dataset contains several types of tumors, gene names, tumor groups and expression levels.

***Tumors***

All the several types of tumors are represented in the alphanumeric format, GSM(XXXX). "GSM" stands for "Global Sample Identifier." GSM is a term commonly used in the Gene Expression Omnibus (GEO) database. Each GSM entry in the database is a different type of tumor obtained from experiment data. For instance, GSM139476 would be a unique identifier for a specific sample in a gene expression experiment stored in the GEO database

***Gene Names***

Gene names are alphanumeric symbols or labels used to identify and refer to specific genes within the context of genetics and genomics. Gene names are essential for communication and data management in molecular biology. The common names given to genes are based on their function, location, or associated phenotype. Gene names can provide insight into the gene's role. Certain genes may be known to play crucial roles in ovarian cancer. For example, BRCA1 and BRCA2 are genes associated with an increased risk of ovarian cancer.

***Tumor Groups***

Endometrioid, serous, mucinous, and clear cell are different subtypes of epithelial ovarian cancer, which is a type of ovarian cancer that originates in the epithelial cells lining the ovaries. These subtypes are characterized by differences in the appearance and behavior of the cancer cells under a microscope These groups are important to consider in the diagnosis and treatment of ovarian cancer because they can impact the choice of treatment strategies and the patient's prognosis. Studying gene expression patterns in groups of genes associated with cancer-related pathways or biological processes can provide insights into the mechanisms of ovarian cancer

***Expression Levels***

The numerical values (e.g., 5362.04 for DDR1 and 309.085 for RFC2) indicate the expression levels of these genes within the group. These values might represent the amount of gene expression (e.g., mRNA levels) in the samples within the group. Expression levels can provide insights into the activity of these genes in the specific condition or group being studied. Changes in gene expression levels are often observed in cancer cells compared to normal cells. Differences in the expression of specific genes or gene signatures can indicate potential abnormalities associated with cancer.

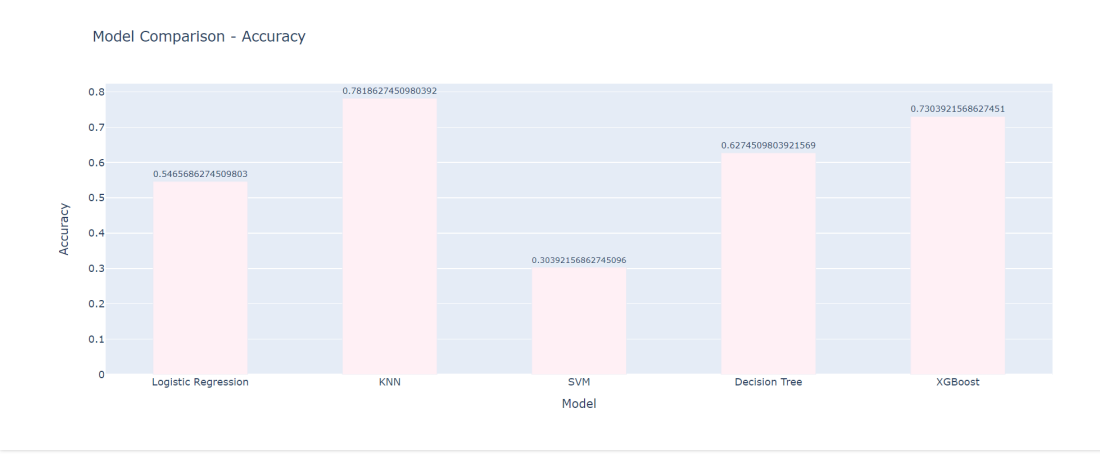
**Data Source**

The following data set from Kaggle has been considered for this project. The link for the same has been provided below.

[Ovarian Dataset](https://www.kaggle.com/datasets/yoshifumimiya/6-ovarian-cancer-datasets/?select=GSE18520.csv)

# **Results and Analysis**

## ***Model Comparison – Accuracy***

* Created an interactive bar chart comparing the accuracies of different machine learning models.
* Model names included in the comparison: Logistic Regression, KNN, SVM, Decision Tree, and XGBoost.
  + 
* Each bar represents the accuracy of a specific model.
  + 
* The chart provides a visual overview of how each model performs in terms of accuracy.
* The accuracy values are displayed on the bars for easy reference.
* The chart helps in identifying which models are more effective for the given task.

## ***Optimal Method***

Based on all the research and testing of the suitable machine learning algorithm for the given data, we found KNN to be the most optimal algorithm considering the overall test accuracy and the F-1 scores in the classification report along with the prediction which can be seen in the confusion matrix.

Below table gives an overview of the comparison used for deciding the best model.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Class** | KNN | | XG Boost | | Decision Tree | | Logistic Regression | | SVM | |
|  | Accuracy | F1 Score | Accuracy | F1 Score | Accuracy | F1 Score | Accuracy | F1 Score | Accuracy | F1 Score |
| **case** | 0.7819 | 0.99 | 0.7206 | 0.93 | 0.5980 | 0.91 | 0.5392 | 0.96 | 0.3064 | 0.31 |
| **Clear cell** | 0.69 | 0.50 | 0.07 | 0.16 | 0.26 |
| **control** | 0.96 | 0.86 | 0.84 | 0.67 | 0.61 |
| **endometrioid** | 0.64 | 0.51 | 0.55 | 0.37 | 0.08 |
| **mucinous** | 0.63 | 0.75 | 0.32 | 0.24 | 0.43 |
| **serous** | 0.79 | 0.76 | 0.72 | 0.79 | 0.28 |

# **Conclusion**

This project aimed at classifying cancer subtypes through an extensive analysis of gene expression data. We meticulously collected, preprocessed, and explored diverse datasets, ensuring data accuracy. Applying Principal Component Analysis (PCA) facilitated dimensionality reduction, and machine learning models, including XGBoost, Decision Trees, Logistic Regression, K-Nearest Neighbors, and Support Vector Machines, were rigorously evaluated.

The models underwent thorough assessment, considering accuracy metrics, classification reports, and confusion matrices. Visualization techniques provided additional insights.

A final model comparison highlighted the strengths of each algorithm, aiding in the selection of the most effective approach for cancer subtype classification. This comprehensive journey ensures a robust foundation for future applications in leveraging gene expression data for cancer classification.

# **Reference**

* Geeks for Geeks, Stack Overflow
* Youtube – Playlists of Stat Quest, Andrew NG, Dhavesh Bhatt & Data Magic to learn about the various needful resources
* Scikit Learn documentation
* Pandas Documentation
* XGBoost Documentation
* OpenAI