UNIVERSITY OF ŽILINA

Faculty of Management Science and Informatics

Data Science Internship Report

Development of Machine Learning Model for Cancer Prediction

During the period – 19th July 2025 to 16th September 2025

Submitted by

Swaroop Mahadeo Ratnaparkhi
Final Year Integrated MSc. Student, Department of Chemistry
Indian Institute of Technology Kharagpur
West Bengal, India – 721302

Under the Guidance of

Dr. Ľuboš Buzna, Professor of Applied Informatics, Faculty of Management Science and Informatics, and Senior Researcher at Department of International Research Projects, University of Žilina, Žilina, Slovakia

Dr. Milan Straka, Researcher at the Department of Mathematical Methods and Operations Research, and Faculty of Management Science and Informatics, University of Žilina, Žilina, Slovakia

Dr. Jozef Kostolný, Researcher at Faculty of Management Science and Informatics, University of Žilina, Žilina, Slovakia

Contents

Data Overview

The dataset is stored in the workbook “Dáta pre bioinformat(1).xlsx”. The Excel workbook contains 9 worksheets titled – “kontrola”, “ Karcinóm endometria_EE”, “EE-nie nádor”, “Kolorektálny karcinóm malígny”, “Kolorektálny adenóm benígny”, “Mechúr bez nádor zmien”, “Mechúr -NIPUC”, “Mechúr IUC” and “Mechúr IUC+NIPUC+prostata”. Each worksheet includes fluorescence intensity measurements of urine samples from patients who either have or do not have a specific type of cancer. The data is summarized in **Table 1**.

Table 1: Overview of the original dataset "Dáta pre bioinformat(1).xlsx"

Group	Characteristics	Samples
kontrola	The group without diagnostic serious chronic diseases with negative strength	83
Karcinóm endometria_EE	Histologically confirmed endometrial tumor	26
EE-nie nádor	Cyst, endometriosis and other non-cancerous changes	14
Kolorektálny karcinóm malígny	Histologically confirmed tumor of various stages and intestinal segments	33
Kolorektálny adenóm benígny	Histologically negative neoplasms	16
Mechúr bez nádor zmien	Histologically undetected tumor - remission or unconfirmed tumor	51
Mechúr -NIPUC	Histologically negative neoplasms	34
Mechúr IUC	Invasive bladder cancer - a set of patients with different types of IUC	24
Mechúr IUC+NIPUC+prostata	Histologically detected various tumor types (NIPUC, IUC, and prostate carcinoma)	1

The data contained some trailing missing columns for the group ‘kontrola.’ Missing values were imputed using linear regression.

Note: From **Table 1** there is only one sample for the class “Mechúr IUC+NIPUC+prostata”, so it was not considered for training machine learning models.

Aim

To develop a machine learning model that can reliably predict whether a person has cancer or not.

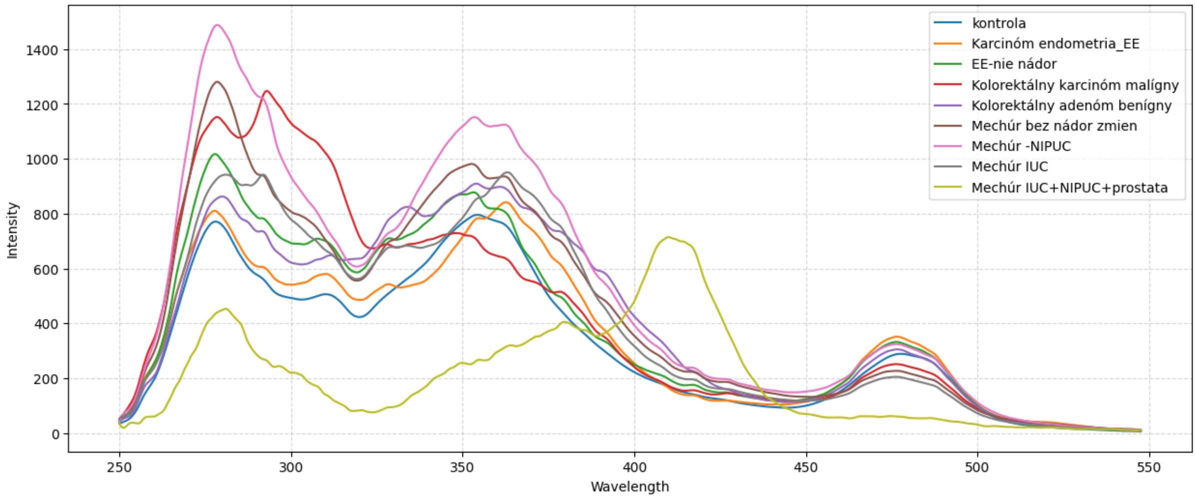
Before applying machine learning algorithms, the data was cleaned by handling missing values and applying Savitzky-Golay smoothing.

Observations

1. Performing Mean Spectra Comparison

Figure 1 shows the mean spectra comparison for all the different classes which gives an idea of the gross similarity between the spectrums of classes.

Figure 1: Mean Spectra Comparison

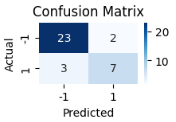
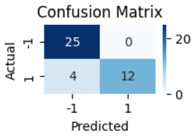
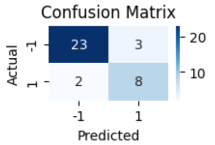


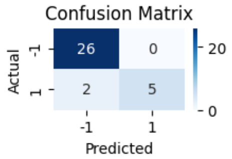
2. Binary Classification Between the Control Group and Different Cancer Groups

The data for control group (labelled as -1) was extracted from "Dáta pre bioinformat(1).xlsx" and paired with the data for each cancer group (labelled as 1).

Subsequently a L1-regularized Logistic Regression model was trained on this data for feature selection and making predictions on the test data. The results are shown in **Table 2**.

Table 2: Binary classification between the pairs of control group (-1) and different cancer group (1) for the model – L1-regularized Logistic Regression

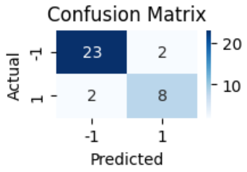
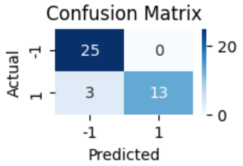
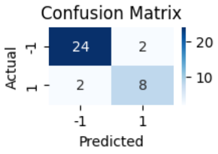
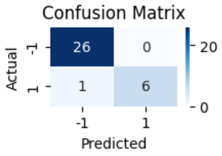
Classification model	Performance on training data – Best cross–validation score (Best weighted f1–score)	Performance on testing data
L1-regularized Logistic Regression for kontrol - Kolorektálny karcinóm malígny	0.888	
L1-regularized Logistic Regression for kontrol - Mechúr bez nádor zmien	0.806	
L1-regularized Logistic Regression for kontrol - Mechúr -NIPUC	0.878	

L1-regularized Logistic Regression for kontrol - Mechúr IUC	0.790	

Remark 1: The L1-regularized Logistic Regression model gives decent performance in distinguishing control group samples from samples belonging to Kolorektálny karcinóm malígny, Mechúr bez nádor zmien, Mechúr -NIPUC, and Mechúr IUC.

A Standard Logistic Regression model was trained on the data corresponding to the selected features obtained from the previous L1-regularized Logistic Regression model. The results are shown in **Table 3**.

Table 3: Binary classification between the pairs of control group (-1) and different cancer groups (1) for the Standard Logistic Regression model

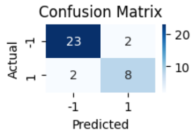
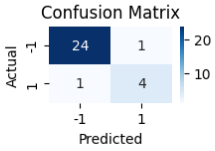
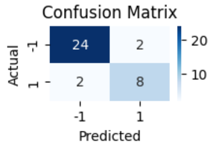
Classification model	Performance on testing data
Standard Logistic Regression for kontrol - Kolorektálny karcinóm malígny	
Standard Logistic Regression for kontrol - Mechúr bez nádor zmien	
Standard Logistic Regression for kontrol - Mechúr -NIPUC	
Standard Logistic Regression for kontrol - Mechúr IUC	

Remark 2: The Standard Logistic Regression model performs slightly **better** than the previous L1-regularized Logistic Regression model in distinguishing control group samples from samples belonging to Kolorektálny karcinóm malígny, Mechúr bez nádor zmien, Mechúr -NIPUC, and Mechúr IUC. **This suggests that a L1-regularized Logistic Regression model trained on the dataset with all**

the features gives poor classification performance as compared to a Standard Logistic Regression model which is trained on the data corresponding to the selected features obtained from the former model.

Similar to the L1-regularized Logistic Regression model a L1-regularized Linear Regression model was trained on the dataset for feature selection and making predictions on the test data. The results are shown in **Table 4**.

Table 4: Binary classification between the pairs of control group (-1) and different cancer group (1) for the model – L1-regularized Linear Regression

Classification model	Performance on testing data
L1-regularized Linear Regression for kontrol - Kolorektálny karcinóm malígny	
L1-regularized Linear Regression for kontrola + Kolorektálny adenóm benígny	
L1-regularized Linear Regression for kontrola + Mechúr -NIPUC	

The L1-regularized Linear Regression model does NOT perform significantly better than the previous Standard Logistic Regression model.

When another Standard Logistic Regression model, with no penalty, was trained on the data corresponding to the selected features obtained from the L1-regularized Linear Regression model, its predictions on the test data did NOT surpass those of the previous Standard Logistic Regression model.

Remark 3: The L1-regularized Linear Regression model does NOT perform effectively in binary classification tasks for either of the following objectives:

- Extracting the important input features
- Making accurate predictions on the test data

The observations are documented in 2025_07_25 → 25_July_2025.ipynb

3. Impact of Oversampling Techniques and Decision Threshold Adjustments

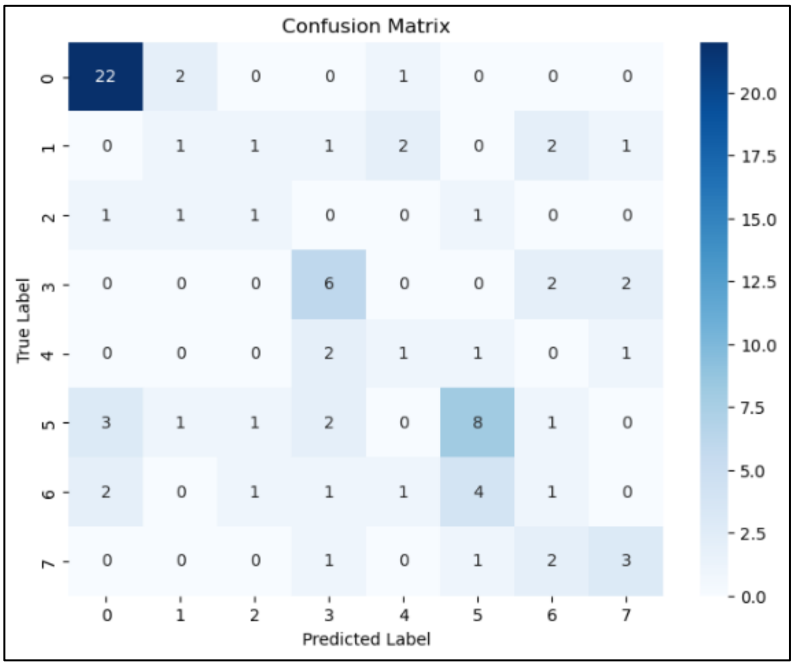
a. Oversampling the minority class in the training data did NOT lead to a significant improvement in the predictive performance of any of the previously described models on the test data.

b. Similarly, adjusting the decision threshold did NOT enhance the performance of any of the models on the test data.

These findings are documented in 2025_08_07 → 'Documenting the observations.xlsx'

4. One-vs-Rest Multiclass Classification Using Linear Regression

The results for one-vs-rest multiclass classification using linear regression models with a tanh-based cost function without regularization:



	0	1	2	3	4	5	6	7
0	22	2	0	0	1	0	0	0
1	0	1	1	1	2	0	2	1
2	1	1	1	0	0	1	0	0
3	0	0	0	6	0	0	2	2
4	0	0	0	2	1	1	0	1
5	3	1	1	2	0	8	1	0
6	2	0	1	1	1	4	1	0
7	0	0	0	1	0	1	2	3

It can be clearly observed that the results for multiclass classification in this case are NOT decent.

The results can be accessed from:
2025_08_07 ->Performing Multiclass classification -> Multiclass_classification_Lin_Reg.ipynb

Note: The code for implementing L1 regularization for this model still needs to be written.

5. Feature Extraction Using Iterative Logistic Regression with L1 Regularization

The Hypothesis:
L1-regularized Logistic Regression can be effectively used for feature extraction. By training L1-regularized Logistic Regression models using a one-vs-rest approach for each class in the dataset, we obtain a distinct set of selected important features from each model. These features from each model are the most decisive in classifying a sample into the corresponding class.

If we take the union of these feature sets across all models, we obtain a comprehensive set that captures the decisive features for all classes. Next, by extracting the data corresponding to this union of features and retraining the L1-regularized Logistic Regression models using the same one-vs-rest strategy, we generate a new union set of selected features. This new set is typically smaller than the previous one but contains more refined and discriminative features.

By repeating this process iteratively, we progressively converge toward a compact set of the most important features for classification.

Following is the step-by-step description of the iterative method:

Step 1: Since the dataset contains 8 distinct output classes (as described in 'Data Overview'), create 8 L1-regularized Logistic Regression models—one for each class. For example, let $f_{class-3}$ be the logistic regression model for 'class-3'. For this model, samples belonging to 'class-3' are labelled as 1, while samples not belonging to 'class-3' are labelled as -1 for training purposes.

Step 2: For each model $f_{class-k}$, perform feature selection and collect the selected features. Take the union of all selected features across the 8 models.

Step 3: Extract the data corresponding to this union of selected features.

Step 4: Repeat the process from Step 1 using the reduced dataset obtained in Step 3.

NOTE: For nuanced details go to:
2025_08_23 -> Articulated Report and Train_Test_Split - Relabelling -> Articulated_Report.pptx

Alert: Implementation Error in Iterative Feature Selection

During my initial implementation of this iterative method, I performed a total of 10 iterations and ended up with 181 selected input features. However, I made a critical mistake. The correct procedure to create $f_{class-k}$ is as follows:

First, divide the original dataset into training and testing sets. Then, convert the output labels into binary form—assigning 1 to samples belonging to $class - k$ and -1 to all other samples. Finally, train the model $f_{class-k}$ using this binary-labelled training data.

Instead of following this procedure, I mistakenly converted the output labels before performing the train-test split. This led to different training and testing samples for each model $f_{class-k}$, which is problematic given the small size of the dataset. As a result, the performance of $f_{class-k_1}$ on its testing data cannot be reliably compared with that of any other model $f_{class-k_2}$, since each model was evaluated on a different test set.

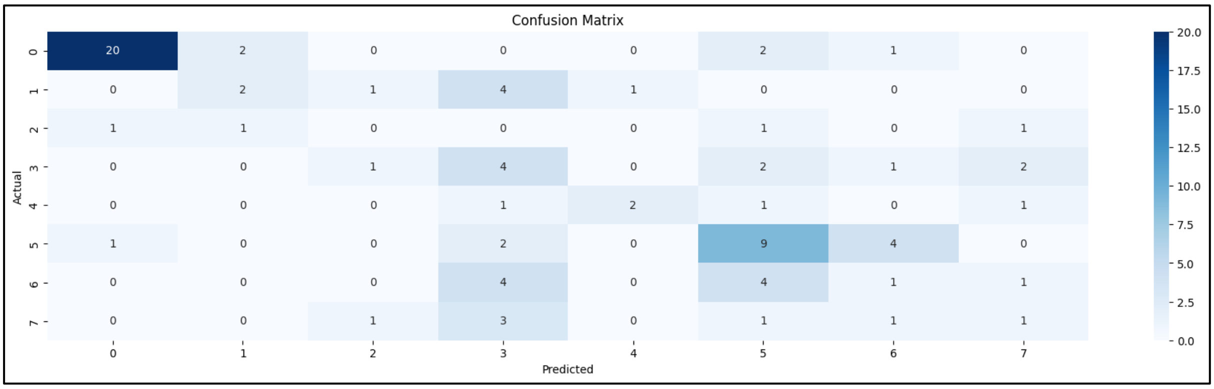
However, here is the catch: Although each model $f_{class-k}$ was trained and tested on different subsets of data, each model is still a valid binary classifier in itself. While direct performance comparisons between models are not meaningful due to inconsistent test sets, the models themselves are not unsuitable for feature extraction. Their validity as binary classifiers ensures that the features they select are still informative for distinguishing $class - k$ from the rest—making them usable in the iterative feature selection process despite the data limitations.

Hence, I proceeded to evaluate various machine learning models using the dataset with 181 input features. Some of the results are presented in subsequent sections.

6. One-vs-Rest Multiclass Classification Using Logistic Regression

<u>For the complete data of 596 input features:</u>

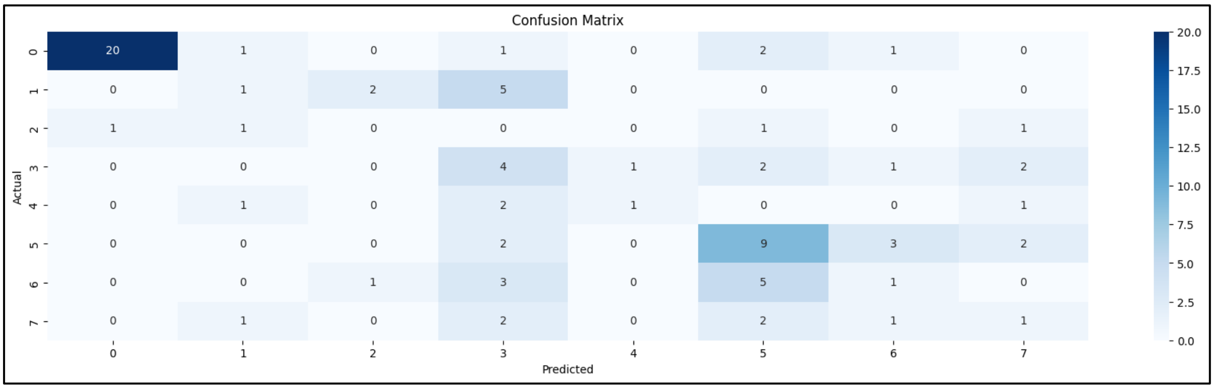
a. Results of performing multiclass classification using a **one vs rest approach** with L1 penalized logistic regression:



Weighted average f1-score = 0.46 and macro average f1-score = 0.34

The results can be accessed from: 2025_08_20 -> Jupyter Notebook Files -> 19_August_2025-4

b. Results of applying PCA (#PCs = 74) to reduce the number of input features and performing multiclass classification using **one-vs-rest approach** with L1-penalized logistic regression:

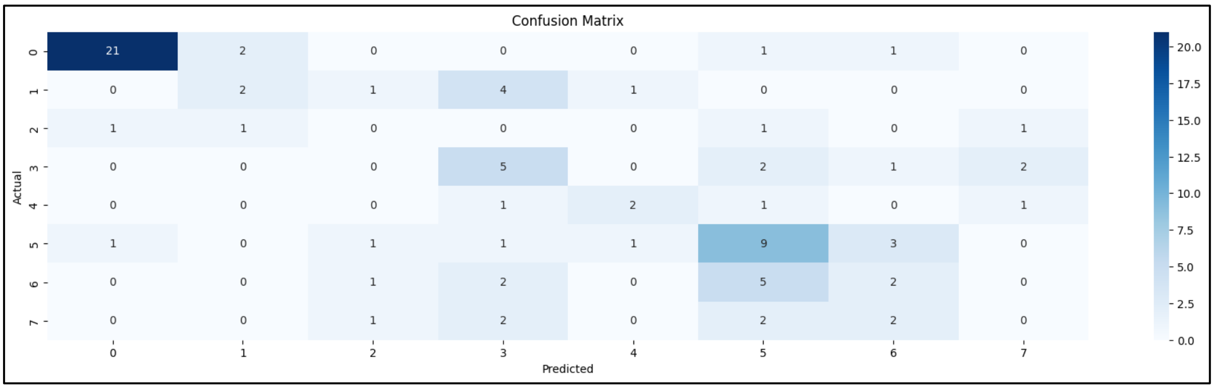


Weighted average f1-score = 0.44 and macro average f1-score = 0.29

The results can be accessed from: 2025_08_20 -> Jupyter Notebook Files -> 19_August_2025-6

For the data with 181 important input features:

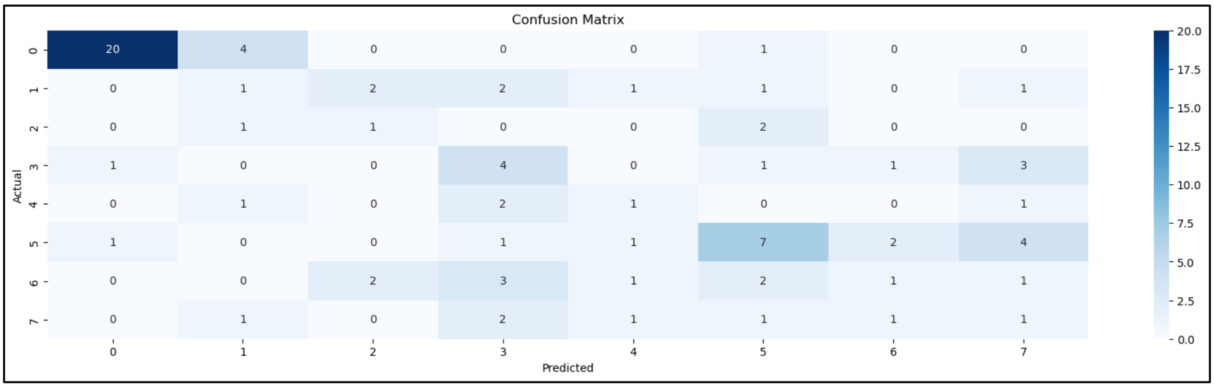
a. Results of performing multiclass classification using a one-vs-rest approach with L1 penalized logistic regression:



Weighted average f1-score = 0.48 and macro average f1-score = 0.34

The results can be accessed from: 2025_08_20 -> Jupyter Notebook Files -> 19_August_2025-9

b. Results of applying PCA (#PCs = 63) to reduce the number of input features and performing multiclass classification using **one-vs-rest approach** with L1-penalized logistic regression:



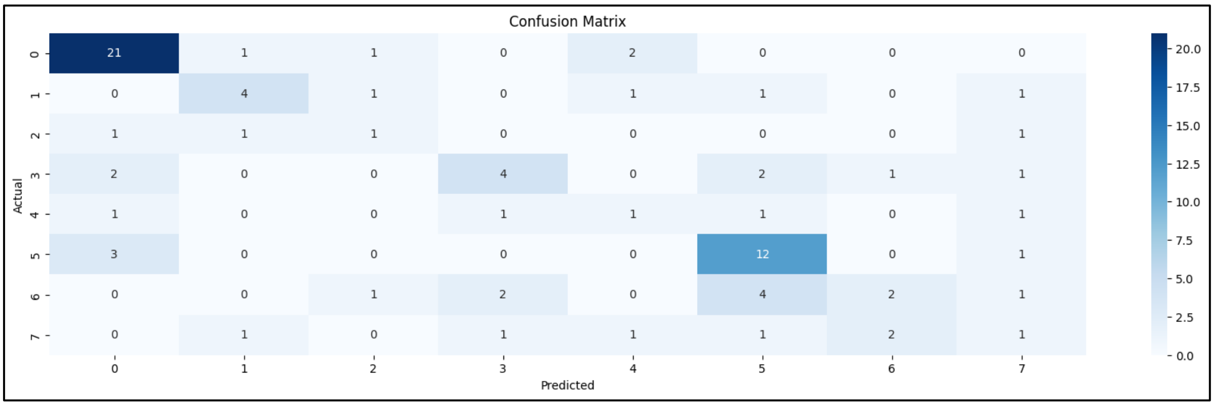
Weighted average f1-score = 0.43 and macro average f1-score = 0.30

The results can be accessed from: 2025_08_20 -> Jupyter Notebook Files -> 19_August_2025-10

Remark 4: The one-vs-rest approach does NOT give decent results.

7. One-vs-One Multiclass Classification Using Logistic Regression

The **best** result after performing multiclass classification using **one-vs-one approach** with L1-penalized logistic regression:



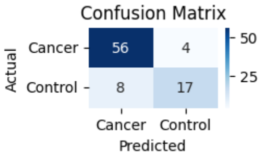
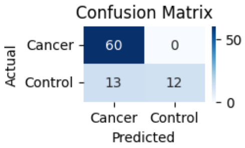
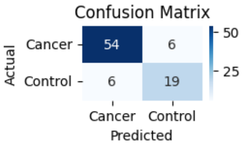
Weighted average f1-score = 0.52 and macro average f1-score = 0.41

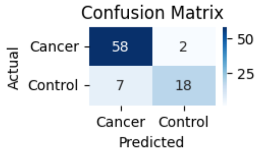
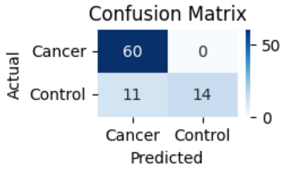
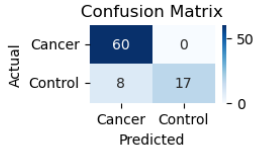
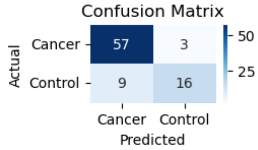
The result can be accessed from: 2025_09_02 -> One vs One Classification.ipynb

Remark 5: Multiclass classification using one vs one approach does not give decent results.

8. Binary classification using trees and ensemble methods

The **best** results after performing binary classification of control vs cancer group using various machine learning models:

Machine Learning model	Performance on the test data
Decision Tree (trained on the data with 182 important input features) PCA was performed	
Random Forest (trained on the data with 182 important input features) PCA was performed	
AdaBoost (trained on the data with 182 important input features) PCA was performed	

Gradient Boosting (trained on the data with 182 important input features) PCA was performed	
XGBoost (trained on the data with 182 important input features) PCA was performed	
CatBoost (trained on data with 596 input features) PCA was performed	
LightGBM (trained on the data with 182 important input features) PCA was performed	

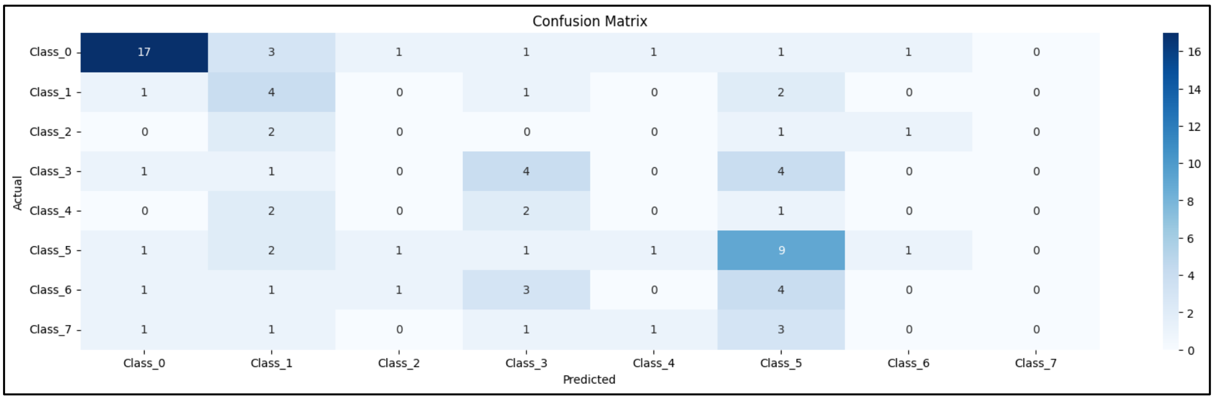
The results can be accessed from: 2025_09_02

It was observed that the models which were trained on the principal components of the data with 182 important input features performed slightly better. **This shows that feature extraction using the iterative L1 regularized logistic regression is helpful in the current context.**

9. Multiclass classification using trees and ensemble methods

The **best** results after performing multiclass classification of control vs cancer group using various machine learning models trained on the data with 596 input features.

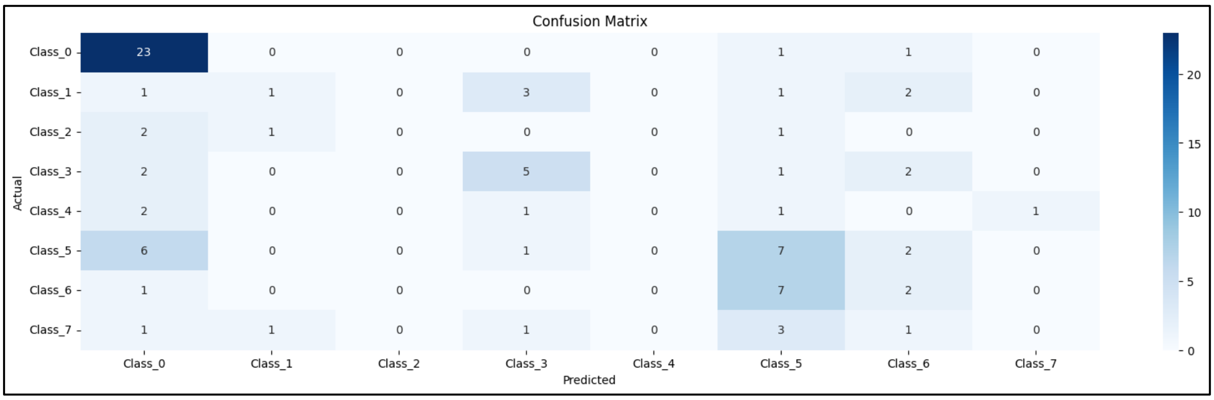
a. Decision Trees



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	17	3	1	1	1	1	1	0
Class_1	1	4	0	1	0	2	0	0
Class_2	0	2	0	0	0	1	1	0
Class_3	1	1	0	4	0	4	0	0
Class_4	0	2	0	2	0	1	0	0
Class_5	1	2	1	1	1	9	1	0
Class_6	1	1	1	3	0	4	0	0
Class_7	1	1	0	1	1	3	0	0

Weighted average f1-score = 0.37 and macro average f1-score = 0.23

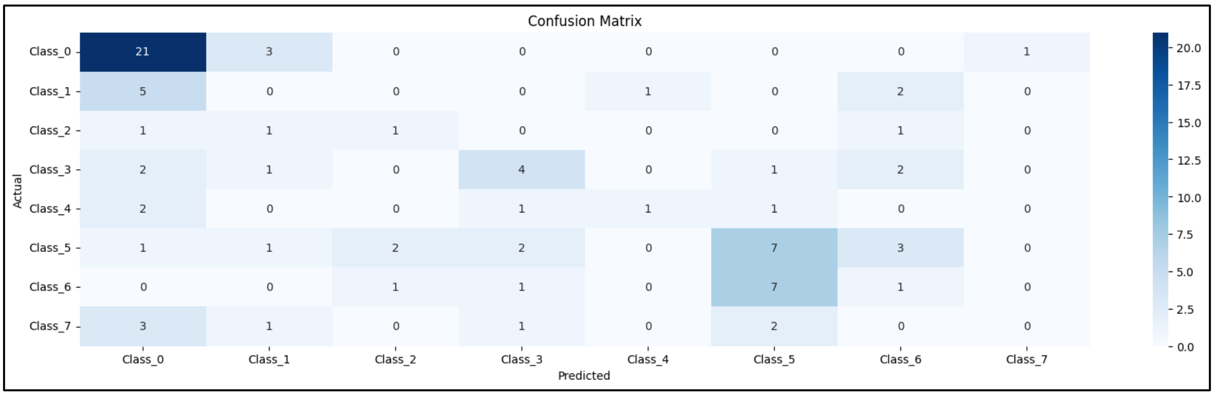
b. Random Forest



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	23	0	0	0	0	1	1	0
Class_1	1	1	0	3	0	1	2	0
Class_2	2	1	0	0	0	1	0	0
Class_3	2	0	0	5	0	1	2	0
Class_4	2	0	0	1	0	1	0	1
Class_5	6	0	0	1	0	7	2	0
Class_6	1	0	0	0	0	7	2	0
Class_7	1	1	0	1	0	3	1	0

Weighted average f1-score = 0.38 and macro average f1-score = 0.24

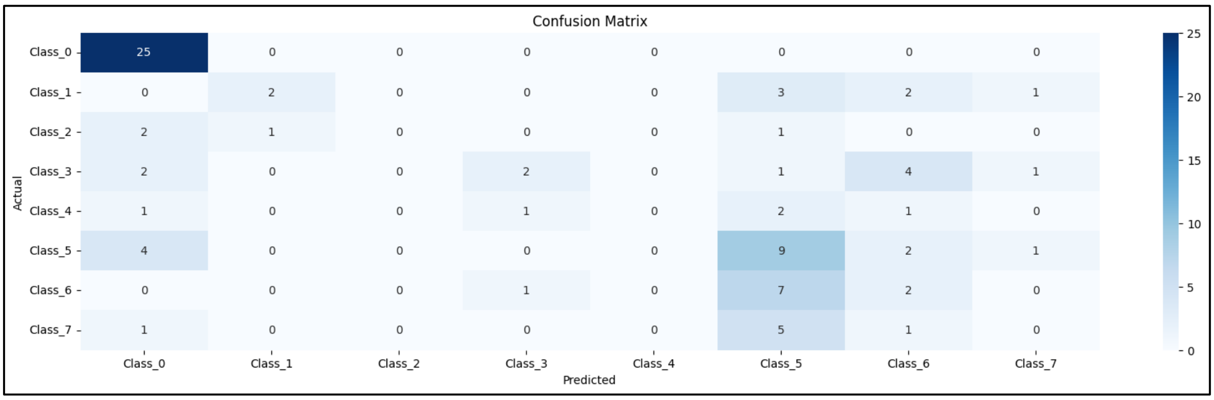
c. AdaBoost



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	21	3	0	0	0	0	0	1
Class_1	5	0	0	0	1	0	2	0
Class_2	1	1	1	0	0	0	1	0
Class_3	2	1	0	4	0	1	2	0
Class_4	2	0	0	1	1	1	0	0
Class_5	1	1	2	2	0	7	3	0
Class_6	0	0	1	1	0	7	1	0
Class_7	3	1	0	1	0	2	0	0

Weighted average f1-score = 0.37 and macro average f1-score = 0.27

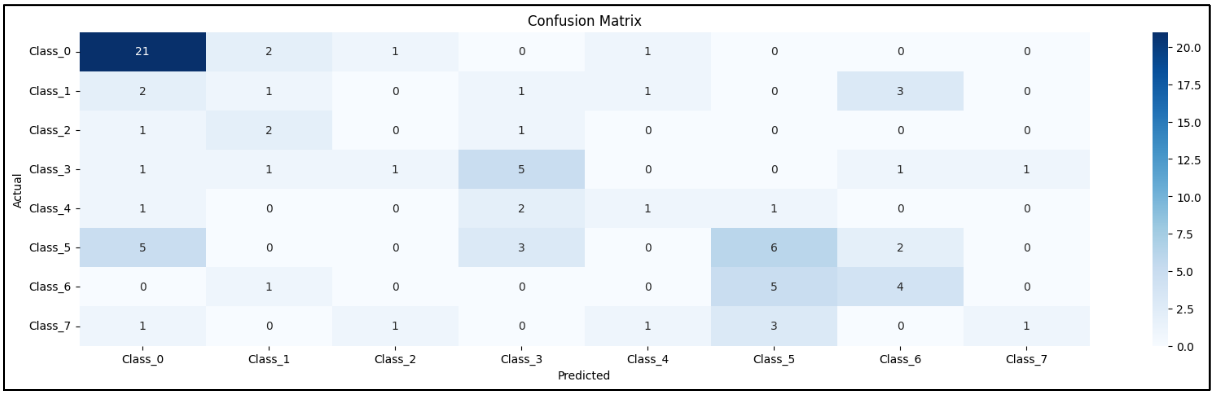
d. Gradient Boosting (I didn’t perform extensive hyperparameter tuning due to high computational cost. With LightGBM this is not the case.)



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	25	0	0	0	0	0	0	0
Class_1	0	2	0	0	0	3	2	1
Class_2	2	1	0	0	0	1	0	0
Class_3	2	0	0	2	0	1	4	1
Class_4	1	0	0	1	0	2	1	0
Class_5	4	0	0	0	0	9	2	1
Class_6	0	0	0	1	0	7	2	0
Class_7	1	0	0	0	0	5	1	0

Weighted average f1-score = 0.41 and macro average f1-score = 0.26

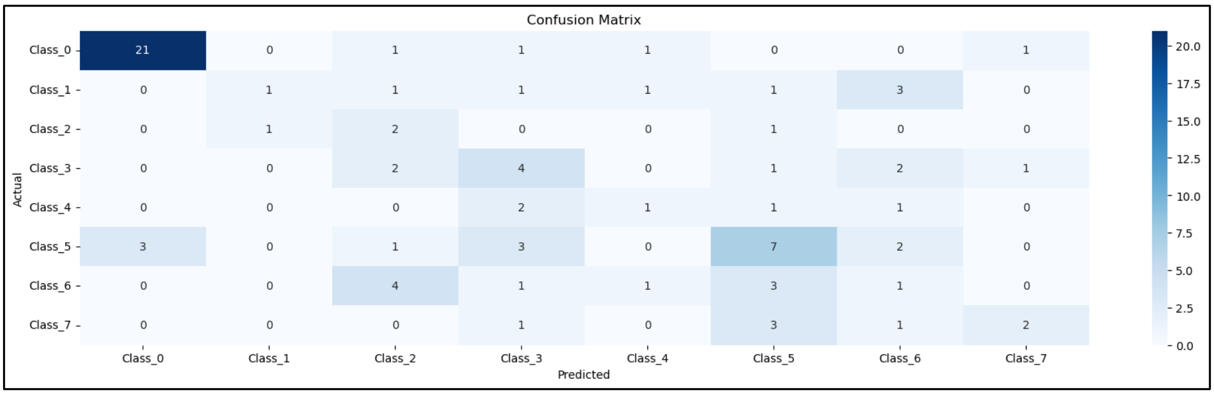
e. XGBoost (I didn’t perform extensive hyperparameter tuning due to high computational cost. With LightGBM this is not the case.)



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	21	2	1	0	1	0	0	0
Class_1	2	1	0	1	1	0	3	0
Class_2	1	2	0	1	0	0	0	0
Class_3	1	1	1	5	0	0	1	1
Class_4	1	0	0	2	1	1	0	0
Class_5	5	0	0	3	0	6	2	0
Class_6	0	1	0	0	0	5	4	0
Class_7	1	0	1	0	1	3	0	1

Weighted average f1-score = 0.43 and macro average f1-score = 0.32

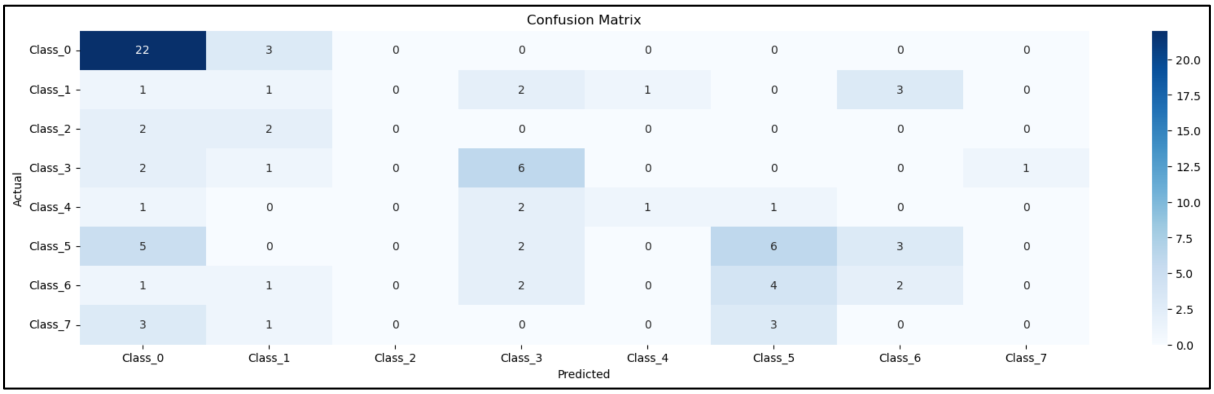
f. Catboost (I didn’t perform extensive hyperparameter tuning due to high computational cost. With LightGBM this is not the case.)



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	21	0	1	1	1	0	0	1
Class_1	0	1	1	1	1	1	3	0
Class_2	0	1	2	0	0	1	0	0
Class_3	0	0	2	4	0	1	2	1
Class_4	0	0	0	2	1	1	1	0
Class_5	3	0	1	3	0	7	2	0
Class_6	0	0	4	1	1	3	1	0
Class_7	0	0	0	1	0	3	1	2

Weighted average f1-score = 0.46 and macro average f1-score = 0.35

g. LightGBM



	Class_0	Class_1	Class_2	Class_3	Class_4	Class_5	Class_6	Class_7
Class_0	22	3	0	0	0	0	0	0
Class_1	1	1	0	2	1	0	3	0
Class_2	2	2	0	0	0	0	0	0
Class_3	2	1	0	6	0	0	0	1
Class_4	1	0	0	2	1	1	0	0
Class_5	5	0	0	2	0	6	3	0
Class_6	1	1	0	2	0	4	2	0
Class_7	3	1	0	0	0	3	0	0

Weighted average f1-score = 0.40 and macro average f1-score = 0.28

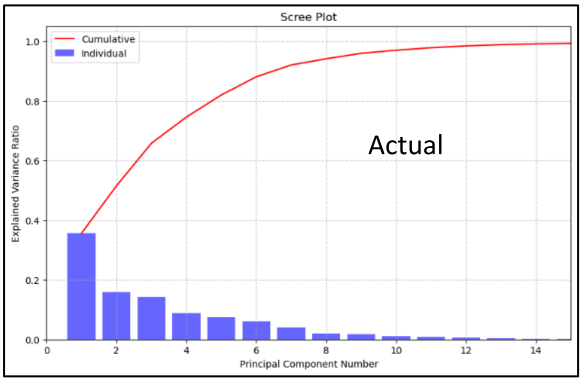
I have not shown the results of trees and ensemble methods trained on the data with 181 important input features because they were not any significantly better.

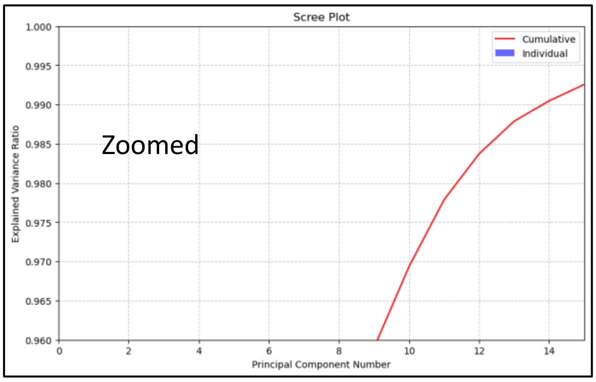
The results can be accessed from: 2025_09_02

Remark 6: The ensemble learning models—AdaBoost, Gradient Boosting, XGBoost, CatBoost, and LightGBM—demonstrate decent performance on binary classification tasks (from Section 7), whereas their performance on multiclass classification (from Section 8) is poor for this dataset.

10. Scree Plot

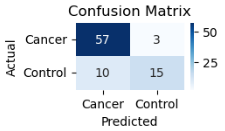
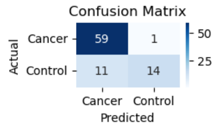
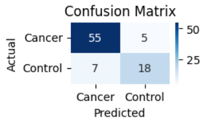
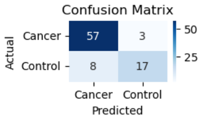
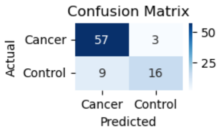
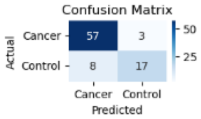
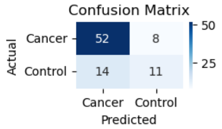
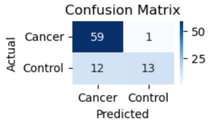
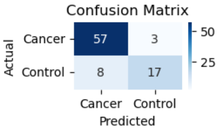
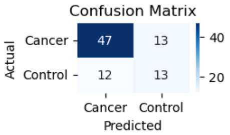
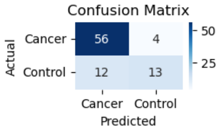
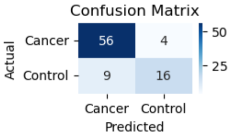
The scree plot for the dataset’s training data is shown below:





The explained variance ratio is above 0.990 at more than 14 principal components.

The results on the test data of several models for binary classification between the control and cancer groups, after varying the number of principal components used for training:

ML model	# PCs = 15	# PCs = 25	# PCs = 33 (explained variance ratio = 0.999)
AdaBoost			
Gradient Boosting			
XGBoost			
LightGBM			

Remark 7: When the ML models are trained on the data with 33 principal components (explained variance = 0.999) the predictions on the test data are better.

Note: In all models before Section 10 ("Scree Plot") that use PCA, the number of principal components was kept at 33 (explained variance >0.999) with the exception of multiclass classification using the one-vs-rest approach in Section 6.

11. Combined ML model

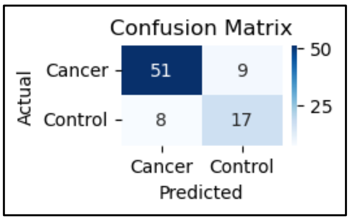
The methodology was adopted from the paper: https://doi.org/10.1039/D2CC03473E

For a better understanding of the method, go to: 2025_09_16 -> PPT.pptx

A brief outline of the method:

Step 1: Train a LightGBM model on the dataset for classification to build multiple trees.
Step 2: For each sample in the training set, record the index of the leaf node it lands on in each tree. These indices represent complex combinations of input features.
Step 3: Convert the leaf node indices into binary vectors using one-hot encoding. This transforms the data into a high-dimensional, sparse format.
Step 4: Use Principal Component Analysis (PCA) to reduce redundancy. Retain components with high variance to preserve meaningful information. This step simplifies the feature space while keeping important patterns.
Step 5: Feed the PCA-transformed features into a logistic regression model or another LightGBM model as the final classifier.
Step 6: Assess model performance using appropriate metrics.

The result for LightGBM + Logistic Regression (l1 penalized) on the data with 596 input features for binary classification between control and cancer group:



Since the research paper doesn’t deals with multiclass classification, it was not performed using this methodology.

The result can be accessed from: 2025_09_16 -> Hybrid Models -> Hybrid Models.ipynb

Conclusion

The findings indicate that using a Standard Logistic Regression model (without regularization), trained on features selected through L1-regularized Logistic Regression, yields improved performance in binary classification tasks. For the current dataset, iterative application of L1-regularized Logistic Regression proved effective for feature selection.

Moreover, binary classification between control and cancer groups consistently outperformed multiclass classification. In particular, the predictions made by XGBoost and CatBoost models in the binary setting (see Section 8 of Observations) demonstrated higher clinical reliability, as they did not misclassify any cancer patient as healthy—a critical requirement in medical diagnostics.

Nonetheless, the limited size of the dataset remains a major constraint.

Additionally, the mean spectra comparison presented in Section 1 of Observations (**Figure 1**) suggests the need for hypothesis testing to determine whether samples from different classes are statistically distinguishable. If the hypothesis test fails to reject the null hypothesis—implying that the samples may originate from the same distribution—then collecting additional data may not significantly enhance classification performance.