DATA CAMP REPORT

Single-Cell RNA-Seq Classification

RAMP Data Challenge on Cell Type Prediction Based on Gene Expression Levels

Students:
Assmaa AL SAMADI
Viviana GAVILANES
Mikael KETEMA

Teacher:
Prof. Nicolas Jouvin
Institution:
University of Paris-Saclay, Évry

December 15, 2023

Contents

1 Introduction

Background on the Data

In this present work, we will provide a detailed exposition of the methodology employed to identifying cell types based on the RNA sequencing data of an individual cell using the Single-cell RNA-seq classification methods. RNA is a molecule that carries the genetic information of cells, and its expression can vary among different cell types or within cells of the same type under different conditions (Alberts et al., 2002).

Single-cell RNA sequencing enables the measurement of gene expression in each individual cell, revealing the heterogeneity and diversity of cells in a biological sample. Numerous methods and tools exist for cell classification based on single-cell RNA sequencing data.

We perfom and test multiple preprocessing and normalisation methods to classify 4 types of cells: NK cells, cancer cells, cytotoxic T lymphocytes cells (CD4+ T) and CD4+ T helper cells. A pairwise gene elimination approach is used to eliminate correlated genes, and TPM, RPKM, FPKM and MinMaxScaler from sklearn as normalisation methods (Zhao et al., 2021). These preprocessings methods was tested for different classifier: KNN,MLP and a combinasion of Random Forest and MLP using Stacking classifier from SKlearn. The study was on 1500 cells splited into train and test with test size $= \frac{1}{3}$. This data challenge uses a small extraction with only 4 cell-types (the labels to predict) from scMARK: Cancer_cells, NK_cells, T_cells_CD4+ y T_cells_CD8+ (Swechha and Mendonca, 2021).

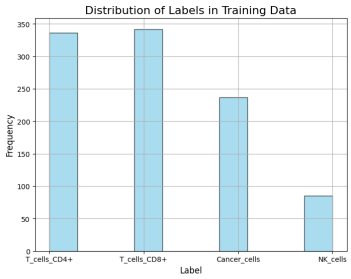
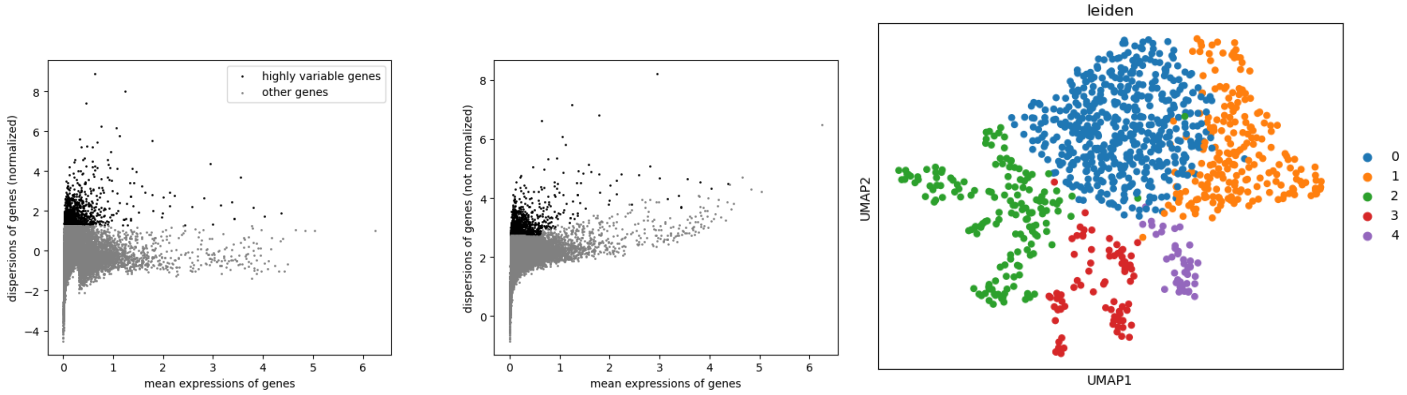


Figure 1: Distribution of samples among the 4 cell-types

2 Data Preprocessing

Data visualization: The "sc.pl.highly_variable_genes" helps to visualize the identification of highly variable genes in the single-cell dataset stored in the anndata object. Highly variable genes are genes whose expression levels vary significantly across individual cells in the dataset. Identifying these genes is crucial for downstream analysis, as they often play key roles in distinguishing different cell types or states.

We use "sc.pl.umap" to create a UMAP (Uniform Manifold Approximation and Projection) plot, which is a dimensionality reduction technique commonly used in scRNA-seq analysis (McInnes et al., 2018). The color=['leiden'] argument indicates that the colors in the UMAP plot should represent the clusters identified by the Leiden algorithm. In our case we found 5 types of cells.



(a) Highly variable genes

(b) UMAP plot

Figure 2: Data visualization

Data Normalization: For data normalization, the total gene length is calculated to compute Transcripts Per Million (TPM) or Reads Per Kilobase Million (RPFKM). However, it was found that all genes have a length of 1.

All normalization values will differ only by a constant of $\frac{X_i}{\sum X_i}$, where X_i denotes reads mapped to the transcript. This approach was used for `T_cells_CD8+` and `NK_cells`. In Addition, MinMaxScaler normalisation from Sklearn is used in some models.

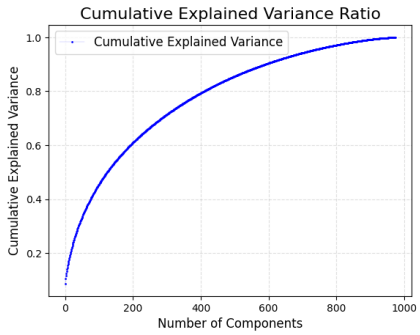


Figure 3: PCA cumulative explained variance ratio after pairwise gene extraction NK and CD8+ cells.

Initial Data Processing: Converted the sparse train input matrix to a dense array and selected data corresponding to specific cell types.

Pairwise gene extraction: The pairwise gene elimination approach consists of eliminate the gene that expressed in 2 specific type of cells. By checking if the the total count for the gene is greater then $\frac{3}{4}$ of the total number of specified genes. Figure 3 depicting the variance captured by different principal components using TruncatedSVD for gene extraction on NK et CD8+ cells.

Filtering: Counted the total number of expressed genes in each cell (`total_cell_counts`). Identified cells with total gene counts exceeding 4000. Removed the filtered cells.

3 Models

In this section, we present the different models used for cell type classification and the experiments conducted. We train The K-nearest Neighbords, Multilayer Perceptron (MLP) and stacking classifier form Sklearn. With these 3 models, different methods of preprocessing and dimensionality reduction is used. To enhance the model's effectiveness, hyperparameter tuning is conducted using RandomizedSearchCV, a method that systematically explores a range of hyperparameter values to identify the optimal configuration. Table 1 summaries the tunned models, methods used and the obtained accuracy on the train and test data set .

Model	**Preprocessing**	**SVD**	**Accuracy Train**	**Accuracy Test**	Train Time
KNN	Pairwise Gene Elimination and MinMAx	100	0.75	0.74	-
KNN	Min-Max Scaling	100	0.796	0.794	15s
MLP	MinMAxScalar	150	0.83	0.81	84s
MLP	Pairwise Gene Elimination NK and CD8+	150	0.86	0.85	38 s
Stacking 1	MinMaxScalar	140	1.00	0.82	73s
Stacking 2	None	None	1.00	0.87	1031s

Table 1: Summary of Model Experiments

For Stacking classifiers, the architectures are as follow:

- Stacking 1 :
  - Base model 1: AdaBoostClassifier.
  - Base model 2: GradientBoostingClassifier
  - Final estimator: RandomForestClassifier.
- Stacking 2 :The stack method is set to 'auto,' allowing the algorithm to dynamically select a method based on the final estimator. Using cv=5 for 5-fold cross-validation enhances collaborative learning between based models to improve overall predictive performance
  - Base model 1: RandomForestClassifier.
  - Base model 2: MLPClassifier (Multilayer Perceptron) with a hidden layer structure of (100, 100)
  - Final estimator: Another Multilayer Perceptron (MLP).

4 Model Performance

In this section, we evaluate the performance of the trained models on the test set. Metrics such as precision, recall, F1-score, and confusion matrices are provided in tables and figures. Precision and Roc Curves Figures 7 show how the prediction for The NK class was better in MLP model with gene extraction compared to then the KNN model Figure 5.

In addition, Figure 10 shows that the stacking classifier has the highest accuracy , but this model has the highest training time compared to others, and with the selected parameters the model doesn't converge.

4.1 KNN Model Performance

4.1.1 Hyperparameter Tuning with Randomized Search

A parameter grid is defined for hyperparameter tuning using randomized search. RandomizedSearchCV is employed to find the best hyperparameters. Detailed performance metrics for the KNN classifier with the best preprocessing method are presented in Table 3.

Parameter	Value
PCA_n_components	100
KNN_weights	distance
KNN_p	2
KNN_n_neighbors	6
KNN	`KNeighborsClassifier()`
Train Accuracy	0.796
Test Accuracy	0.794

Table 2: Best Hyperparameters and Accuracy

	precision	recall	f1-score	support
Cancer_cells	1.00	0.94	0.97	118.0
NK_cells	0.80	0.47	0.59	43.0
T_cells_CD4+	0.83	0.78	0.80	168.0
T_cells_CD8+	0.69	0.84	0.76	171.0
accuracy	0.81		0.81	500
macro avg	0.83	0.76	0.78	500
weighted avg	0.82	0.81	0.81	500

Table 3: Report Metrics Tuned with Min-Max

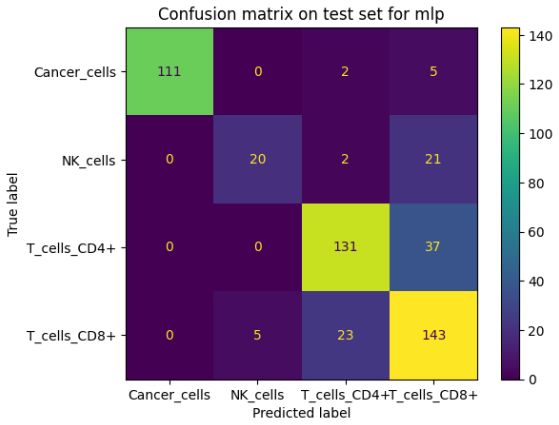


Figure 4: Accuracy table

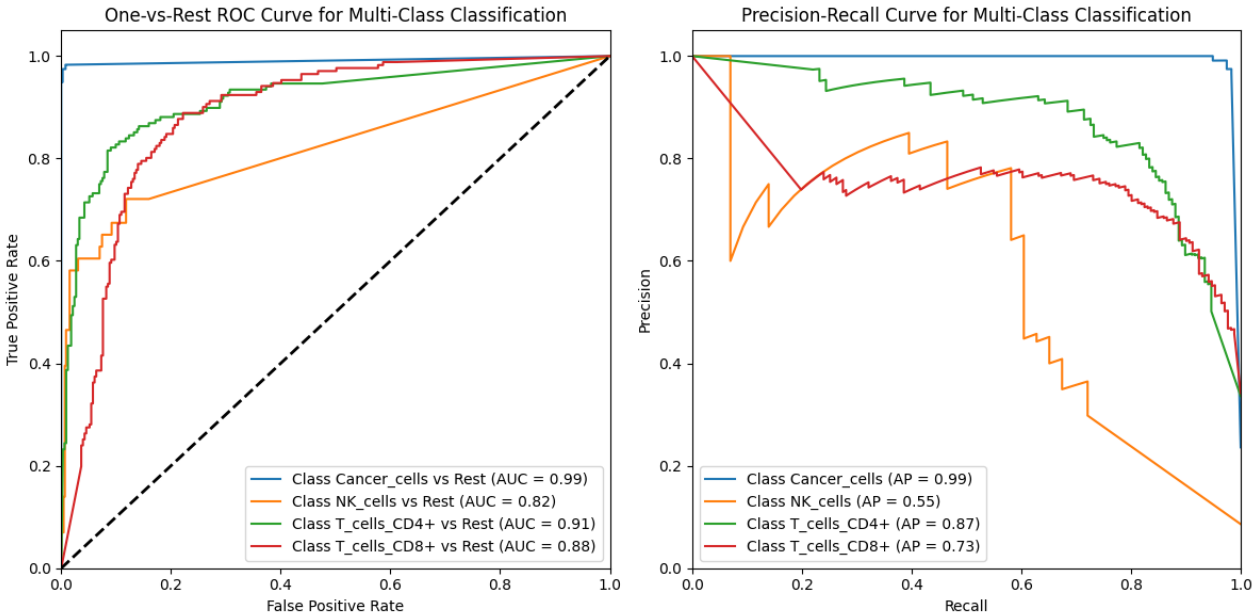


Figure 5: ROC and Precision-Recall Curves for KNN Classifier tunned with Min-Max

4.1.2 KNN tuned with pairwise gene extraction

Parameter	Value
PCA_n_components	120
KNN_weights	distance
KNN_p	2
KNN_n_neighbors	6
KNN	`KNeighborsClassifier()`
Train Accuracy	0.75
Test Accuracy	0.74

Table 4: Best Hyperparameters and Accuracy

	precision	recall	f1-score	support
Cancer_cells	1.00	0.92	0.96	118.0
NK_cells	0.89	0.40	0.55	43.0
T_cells_CD4+	0.66	0.81	0.73	168.0
T_cells_CD8+	0.66	0.64	0.65	171.0
accuracy			0.74	500
macro avg	0.80	0.69	0.72	500
weighted avg	0.76	0.74	0.74	500

Table 5: Report Metrics

4.2 MLP Model Performance

Similar performance metrics for the MLP classifier are provided in a separate table. Detailed performance metrics for the MLP with the best SVD method are presented in Table 9.

4.2.1 MLP with gene extraction

A parameter grid is defined for hyperparameter tuning using randomized search. RandomizedSearchCV is employed to find the best hyperparameters.

Parameter	Value
mlp__solver	adam
mlp__max_iter	600
mlp__hidden_layer_sizes	(2000,150)
mlp__alpha	0.001
mlp__activation	tanh
PCA__n_components	150
Train Accuracy	0.86
Test Accuracy	0.85

Table 6: Best Hyperparameters and Accuracy for MLP Tuned

	precision	recall	f1-score	support
Cancer_cells	0.99	0.98	0.99	118.0
NK_cells	0.74	0.65	0.69	43.0
T_cells_CD4+	0.82	0.89	0.85	168.0
T_cells_CD8+	0.82	0.78	0.80	171.0
accuracy			0.85	500
macro avg	0.84	0.83	0.83	500
weighted avg	0.85	0.85	0.85	500

Table 7: Report Metrics

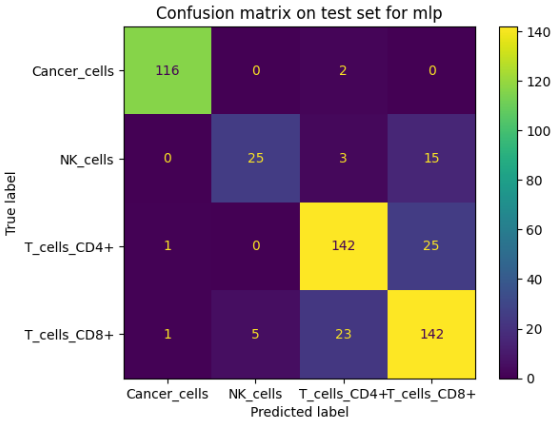


Figure 6: Accuracy table for MLP tuned with SVD

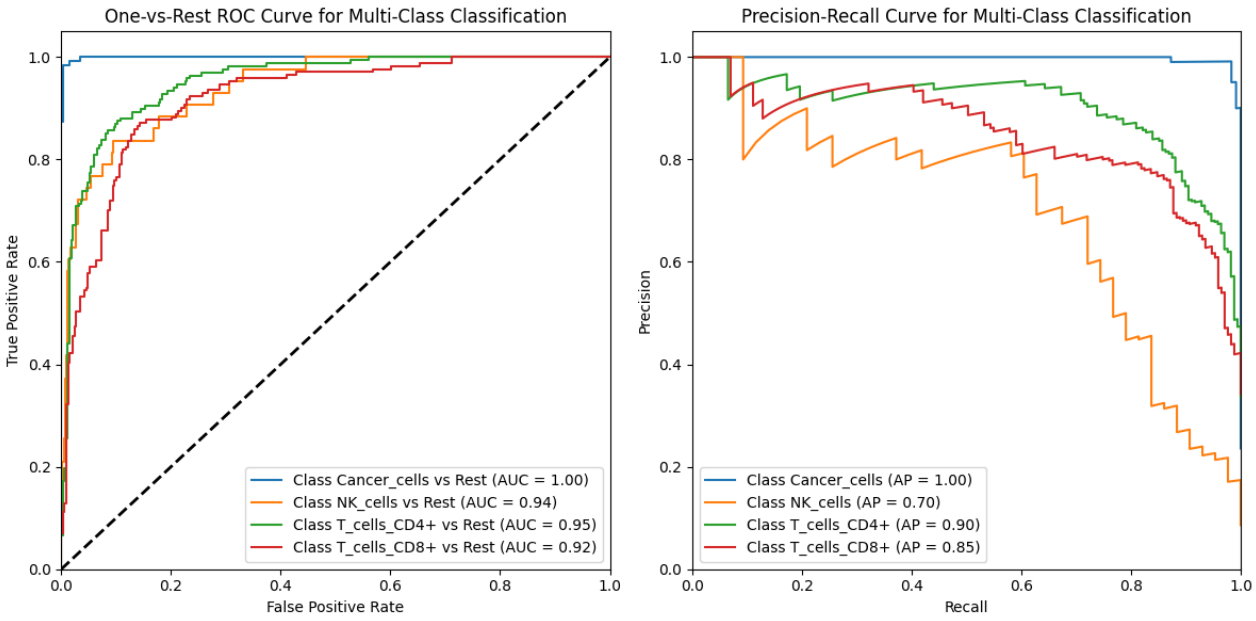


Figure 7: ROC and Precision-Recall Curves

4.2.2 MLP with MinMaxScaler

Parameter	Value
mlp__solver	lbfgs
mlp__max_iter	500
mlp__hidden_layer_sizes	(500, 100)
mlp__alpha	0.01
mlp__activation	tanh
PCA__n_components	120
Train Accuracy	0.83
Test Accuracy	0.81

Table 8: Best Hyperparameters and Accuracy

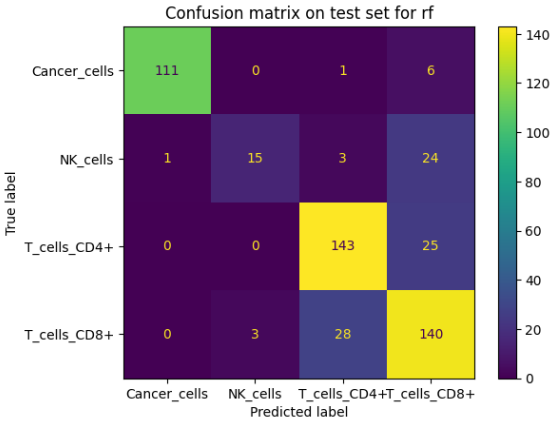
	precision	recall	f1-score	support
Cancer_cells	1.00	0.94	0.97	118.0
NK_cells	0.80	0.47	0.59	43.0
T_cells_CD4+	0.83	0.78	0.80	168.0
T_cells_CD8+	0.69	0.84	0.76	171.0
accuracy			0.81	500
macro avg	0.83	0.76	0.78	500
weighted avg	0.82	0.81	0.81	500

Table 9: Report Metrics

4.3 Stacking 1

	precision	recall	f1-score	support
Cancer_cells	0.99	0.94	0.97	118.0
NK_cells	0.83	0.35	0.49	43.0
T_cells_CD4+	0.82	0.85	0.83	168.0
T_cells_CD8+	0.72	0.82	0.77	171.0
accuracy			0.82	500
macro avg	0.84	0.74	0.76	500
weighted avg	0.83	0.82	0.81	500

(a) Report Metrics



(b) Accuracy Table

Figure 8: Metrics and Accuracy Table

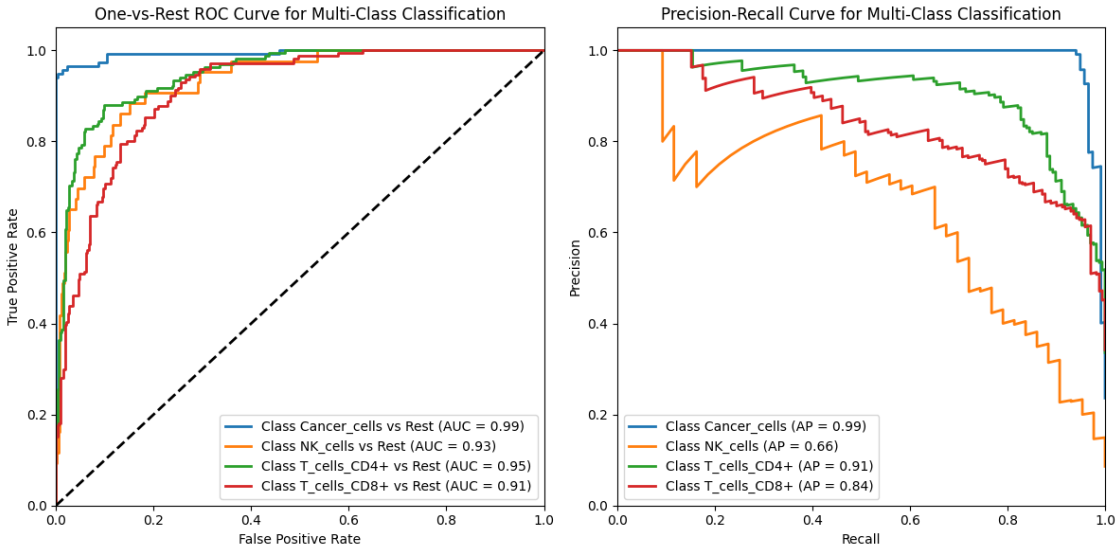
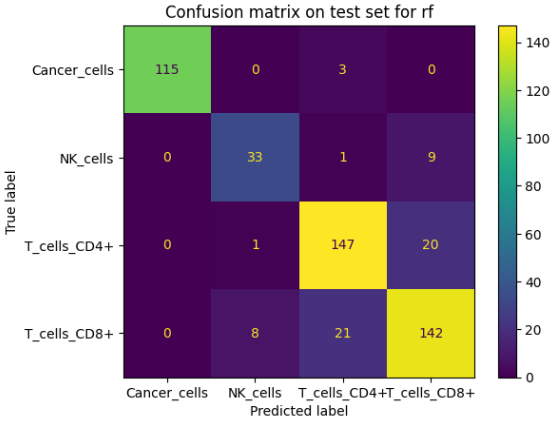


Figure 9: ROC and Precision-Recall Curves

4.4 Stacking 2 : Model Construction with MLP

	precision	recall	f1-score	support
Cancer_cells	1.00	0.97	0.99	118.0
NK_cells	0.79	0.77	0.78	43.0
T_cells_CD4+	0.85	0.88	0.86	168.0
T_cells_CD8+	0.83	0.83	0.83	171.0
accuracy			0.87	500
macro avg	0.87	0.86	0.86	500
weighted avg	0.87	0.87	0.87	500

(a) Report Metrics



(b) Accuracy Table

Figure 10: Metrics and Accuracy Table

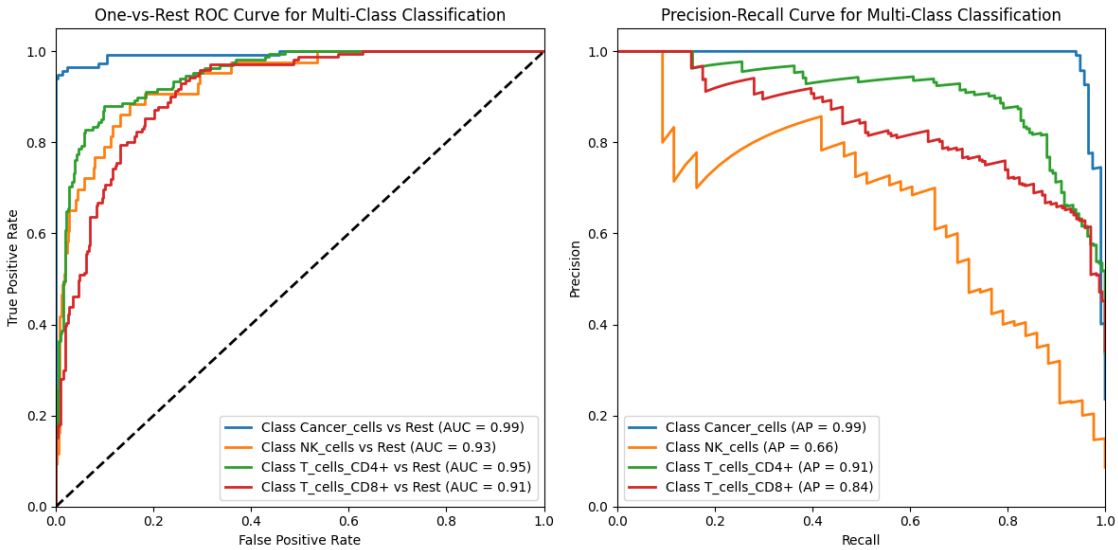


Figure 11: ROC and Precision-Recall Curves

5 Conclusion

The training of multiple model shows that the accuracy changes with the change of normalization and the preprocessing methods. To handle bad accuracy, pairwise gene elimination approach is used with MLPClassifier. The Experiments using the pairwise gene elimination show that the model performance enhanced only by eliminating the NK and CD8+ cells. From this we can conclude that there exist genes expressed in CD8+, CD4-T and cancer cells and the elimination of these genes give bad accuracy. Moreover the training a stacking classifier with simple normalization, gives a good accuracy. The weak point of the former is its complexity and their high number of parameters. All models were unable to classify well the NK cells, where in the training data this class has the lowest representation. Which is expected in machine learning classification task with imbalanced data. To classify such data, we need to find the right normalization method to combine it with gene extraction and fit it to a simple classifier. And if its possible train the Model on more samples since the training set is only 1000 cells.

References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. (2002). *Molecular Biology of the Cell*. Garland Science.

McInnes, L., Healy, J. and Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv preprint arXiv:1802.03426.

Swechha and Mendonca (2021). *scMARK: an 'MNIST'-like benchmark to evaluate and optimize models for unifying scRNA data*.

Zhao, Y., Li, M., Konaté, M. et al. (2021). *TPM, FPKM, or Normalized Counts?* Journal of Translational Medicine *19*, 269.