NORDIC MACHINE INTELLIGENCE

Nordic Machine Intelligence, vol. X, pp. 1–8, 2025
Received 12 May 2025 / published Pending
https://doi.org/10.5617/nmi.some-doi

Methodology for Gene Expression Analysis in Colon Cancer: Biomarker Identification and Pathway Analysis

Abdessamad Hnioua[1], Oumayma Banouar[2]

1. *Master IAII, Faculty of Sciences and Technology, Cadi Ayyad University, Marrakech, Morocco*
2. *Dr., Faculty of Sciences and Technology, Cadi Ayyad University, Marrakech, Morocco*

Abstract

This study addresses the challenge of distinguishing between **normal** and **tumor tissues** using **gene expression data** related to **colon cancer**. By applying **machine learning** classification techniques, we aim to identify a **minimal gene signature** that enables accurate classification while reducing the dataset's **dimensionality**. Our methodology involved rigorous **data preprocessing**, **feature selection** based on a **correlation threshold** of 0.70, and **5-fold cross-validation** to ensure model reliability. The **Random Forest classifier** outperformed other models, achieving high accuracy with only **three selected genes** (**RNF43**, **DAO**, **UGP2**). In contrast, **Logistic Regression** and **SVM** required five genes for comparable performance. **Random oversampling** was used to address **class imbalance**, improving model robustness without introducing complexity. Performance evaluation using metrics such as **precision**, **recall**, **F-score**, and **AUC** confirmed the validity of the selected approach. Our findings highlight the effectiveness of **ensemble learning** and suggest the potential of these genes as **biomarkers** for **colon cancer diagnostics**.

Keywords: machine learning; supervised learning; gene expression; colon cancer; classification; feature selection; biomarkers; ensemble methods; logistic regression; support vector machine; random forest; cross-validation; diagnostic prediction;model evaluation; precision; random oversampling.

Introduction

The identification of **critical biomarkers** and the understanding of the mechanisms underlying **cancer biology** are central challenges in the field of **bioinformatics**. With the growing availability of **large-scale genomic data**, especially **gene expression** datasets, the task has become more achievable, yet remains highly complex due to the **high dimensionality** and variability of the data. In cancer research, gene expression analysis serves as a powerful tool to gain insights into **tumor characteristics**, predict **disease outcomes**, and guide **treatment decisions**.

This study specifically focuses on analyzing gene expression data derived from the **"Gene Expression of Colon Cancer"** dataset available on **Kaggle**. The dataset consists of **804 tissue samples**, each described by the expression levels of **60 genes**. Additionally, it includes a binary target variable, `tissue_status`, which classifies each sample as either **normal** or **tumoral**. By analyzing this dataset, we aim to deepen our understanding of the **molecular features** that distinguish **cancerous** from **non-cancerous tissues**.

The overarching goal of this study is to explore and evaluate various **machine learning** techniques for the **prediction** of tissue status based on gene expression profiles. This involves a structured workflow comprising **data preprocessing**, **feature selection**, **model training**, and **performance evaluation**. A key component of the study is to identify the most **relevant genes** that contribute significantly to classification accuracy, while simultaneously reducing the dataset's **computational complexity** and enhancing model **efficiency**.

This investigation will serve both as a methodological benchmark and a biological insight study, ultimately contributing to the development of more accurate and interpretable models for **colon cancer diagnosis**.

Methodology

Data Preparation

The "Gene Expression of Colon Cancer" dataset, sourced from Kaggle, comprises 804 tissue samples, each characterized by the expression levels of 60 genes (e.g., `ADH1C`, `CTSS`) and a binary target variable, `tissue_status` (normal or tumoral). The dataset includes a unique identifier (`id_sample`), which was excluded from analysis.

Gene Dataset Overview Table presents a comprehensive compilation of 60 genes analyzed across 804 non-null samples, with each gene characterized by its key functional attributes. This diverse gene set encompasses multiple biological domains, including metabolic enzymes (`ADH1C`, `UGP2`, `FABP1`), cellular signaling regulators (`KLF10`, `ERRFI1`, `RNF43`), membrane transporters (`SLC7A5`, `SIDT1`), transcription factors (`FOXF2`, `SOX18`), structural proteins (`TPM1`, `NCAPH`), and immune modulators (`CTSS`, `IGLV8-61`). Notably, several tumor suppressors (`PRUNE2`, `SAMD9`, `PLAAT4`) and oncogenic pathway components are represented, suggesting potential relevance to cancer biology. The expression profiles of these genes, measured as floating-point values, provide insight into molecular mechanisms underlying various physiological and pathological conditions, including metabolic disorders, inflammatory diseases, and malignancies. This gene panel thus constitutes a valuable resource for investigating cellular functions and disease processes at the molecular level.

The distribution of `tissue_status` is perfectly balanced, with 402 samples (50%) labeled as normal and 402 samples (50%) labeled as tumoral, as visualized in Figure 4. This balance ensures unbiased model training, as both classes are equally represented.

No missing values were detected in the dataset (`Nombre de valeurs manquantes: 0`), allowing all samples to be included without the need for imputation. This completeness ensures the integrity and representativeness of the data throughout the analysis.

To prepare the gene expression data for machine learning models, **z-score standardization** was applied. Standardization transforms each feature to have a mean of 0 and a standard deviation of 1. This step is crucial for models sensitive to the scale of input features (e.g., logistic regression, SVM, KNN), as it ensures that all genes contribute equally during the learning process, avoiding dominance by features with larger ranges.

Data Balancing with Oversampling

After dividing the dataset into **training** and **testing sets**, we applied an **oversampling technique** to ensure **class balance** in the training data. Specifically, we implemented **Random Oversampling**, which resulted in **perfectly balanced class distribution** for model training. The resampling process successfully achieved **equal representation** of both tissue types (**302 normal** and **302 tumoral** samples) in the training dataset, as visualized in

Table 1: Complete list of genes with functional descriptions.

Gene	Description
ADH1C	Alcohol dehydrogenase involved in ethanol metabolism.
DHRS11	Dehydrogenase/reductase involved in steroid metabolism.
UGP2	Enzyme in carbohydrate metabolism (UDP-glucose pyrophosphorylase).
SLC7A5	Amino acid transporter, important in cancer cell growth.
CTSS	Protease involved in antigen processing and tumor invasion.
DAO	Degrades histamine and other amines, modulates inflammation.
NIBAN1	Stress response gene, implicated in tumor progression.
PRUNE2	Tumor suppressor candidate, regulates cell proliferation.
FOXF2	Transcription factor regulating epithelial-mesenchymal transition.
TENT5C	Poly(A) polymerase involved in RNA stability.
KLF10	Transcription factor involved in TGF-beta signaling.
FABP1	Fatty acid-binding protein, role in lipid metabolism.
RPSAP19	Pseudogene associated with ribosomal protein regulation.
NCAPH	Component of condensin complex, important in chromosome structure.
TPM1	Tropomyosin protein involved in cytoskeleton regulation.
PLA2G12B	Secretory phospholipase, involved in lipid metabolism.
PLAAT4	Enzyme with tumor suppressor activity.
IGLV8-61	Immunoglobulin variable region, role in immune response.
GSS	Glutathione synthesis enzyme, important for oxidative stress defense.
L1TD1	Stem cell marker, involved in mRNA regulation.
RNF186	RING finger E3 ubiquitin ligase, associated with inflammatory bowel disease.
HES2	Transcriptional repressor in Notch signaling pathway, regulates cell differentiation.
MXRA8	Matrix remodeling protein, receptor for arthritogenic alphaviruses.
SOX18	Transcription factor involved in vascular development and lymphangiogenesis.
NDFIP2	Adaptor protein involved in protein ubiquitination and EGFR trafficking.
SIAE	Sialic acid acetylesterase, regulates B-cell antigen receptor signaling.
NEURL1B	E3 ubiquitin ligase, plays role in Notch signaling regulation.
DDIT4	DNA damage-inducible transcript 4, negative regulator of mTOR pathway.
TRPM4	Calcium-activated ion channel, regulates calcium oscillations.
RETREG1	Reticulophagy regulator 1, involved in endoplasmic reticulum autophagy.
OTULINL	OTU deubiquitinase with linear linkage specificity like, regulates NF-kB signaling.
CPVL	Carboxypeptidase, vitellogenic-like, involved in protein processing.

SAMD9	Sterile alpha motif domain-containing protein 9, tumor suppressor.
EPN3	Epsin 3, involved in clathrin-mediated endocytosis.
CRYBG2	Beta-gamma crystallin domain-containing protein 2, role in lens development.
GIPC2	GIPC PDZ domain-containing family member 2, scaffold protein in cell signaling.
P3H2	Prolyl 3-hydroxylase 2, collagen modification enzyme.
STEAP3	Metalloreductase, involved in iron homeostasis and vesicle trafficking.
THNSL2	Threonine synthase-like 2, amino acid metabolism enzyme.
TRAPPC14	Transport protein particle complex subunit 14, membrane trafficking.
RHBDL2	Rhomboid-like protein 2, intramembrane serine protease.
RPP25	Ribonuclease P protein subunit p25, tRNA processing.
SEMA4C	Semaphorin 4C, axon guidance and cell migration regulator.
RNF43	RING finger protein 43, negative regulator of Wnt signaling.
EPS8L1	EPS8-like protein 1, regulates actin cytoskeleton dynamics.
TOR4A	Torsin family 4 member A, ATPase involved in protein folding.
PAQR5	Progestin and adipoQ receptor family member 5, membrane progesterone receptor.
SIDT1	SID1 transmembrane family member 1, RNA transporter.
ESRP1	Epithelial splicing regulatory protein 1, regulates alternative splicing.
SYTL2	Synaptotagmin-like protein 2, involved in vesicle trafficking.
BSPRY	B-box and SPRY domain-containing protein, regulates calcium channels.
CDHR2	Cadherin-related family member 2, cell adhesion molecule.
ERRFI1	ERBB receptor feedback inhibitor 1, negative regulator of EGFR signaling.
CLIC5	Chloride intracellular channel protein 5, maintains cytoskeleton structure.
PLLP	Plasmolipin, myelin protein involved in ion transport.
GAL	Galanin, neuropeptide involved in various physiological functions.
CRYL1	Crystallin lambda 1, metabolic enzyme involved in sugar metabolism.
YBX2	Y-box binding protein 2, involved in mRNA storage and stability.
ANGPTL4	Angiopoietin-like 4, regulator of lipid metabolism and angiogenesis.

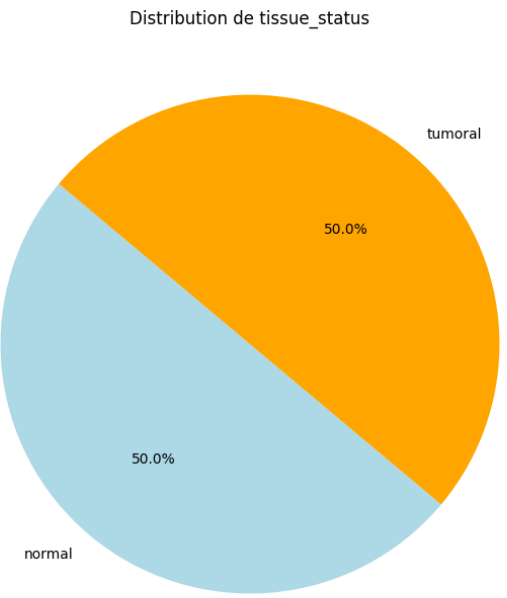

Figure 1: Pie chart illustrating the balanced distribution of tissue status in the Gene Expression of Colon Cancer dataset, with 50% normal and 50% tumoral samples (402 samples each).

Figure 4.

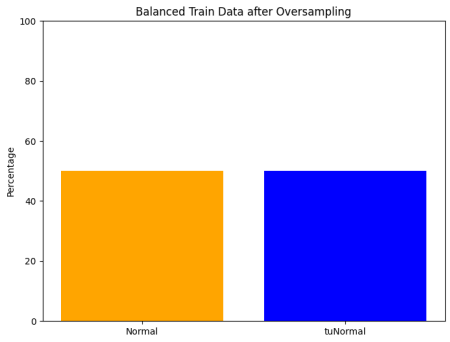

Figure 2: Bar chart illustrating the **balanced distribution** of tissue samples in the training dataset after applying **random oversampling**, with equal proportions (**50%**) of **normal** and **tumoral** samples.

While the **SMOTE** (*Synthetic Minority Oversampling Technique*) algorithm was considered as an alternative approach, the current implementation utilized **Random Oversampling** for its **simplicity** and **effectiveness**. The mathematical formulation of **SMOTE**, which generates **synthetic samples**, is given by:

$$x_{new} = x_i + \lambda \cdot (\hat{x}_i - x_i) \tag{1}$$

where x_i is a sample from the **minority class**, $\hat{x}_i$ is one of its k **nearest neighbors**, and $\lambda \in [0, 1]$ is a **random number**. This **balanced representation** ensures that the model receives **equal exposure** to both tissue types during training, improving the **reliability** of subsequent **gene expression analysis**.

Proposed Approach

In this scientific paper, my approach focuses on selecting the **most impactful features** (genes) while maintaining **predictive performance**, with the goal of **reducing costs** using a set of 60 genes. My objective is to train **classification models** and ultimately propose a **predictive model** based on a **reduced number of genes** while preserving comparable performance.

To achieve this, I begin by applying a **feature selection method**. I then determine the **minimum number of genes** that maximizes accuracy. To avoid **redundancy**, I eliminate less impactful genes when they are **highly correlated** (with a threshold set at 0.70 in absolute value) with a more important gene. If **accuracy decreases** after this elimination, I increase the gene set size by adding the next most impactful gene, and repeat this **loop** until the model achieves **stable performance** equivalent to that obtained with 60 genes.

Subsequently, I **tune and improve** the classification models to obtain the best possible results. Finally, I **evaluate** the models using several metrics: **recall**, **ROC curve**, **precision**, **F-score**, and **confusion matrix**. Given the relatively small size of the dataset (803 samples), I also display **cumulative confusion matrices** obtained through **cross-validation** for more robust evaluation.

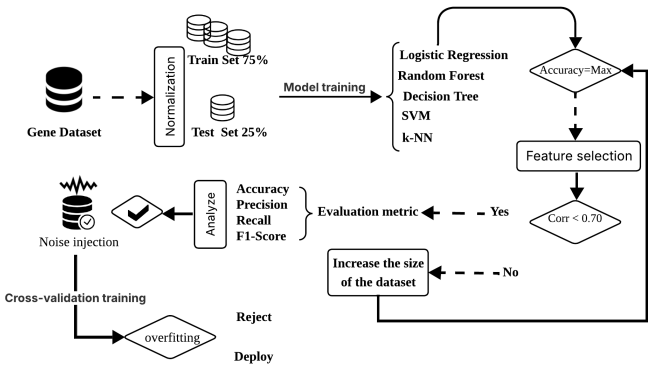

Figure 3: Pipeline for Cost-Effective Gene Selection, Model Training, and Evaluation.

Methods for Estimating Gene Importance using Classification Models

In this study, we employed several classification algorithms to estimate gene importance. For each model, genes are ranked according to the importance assigned by the model, based on specific criteria detailed below.

Logistic Regression:
The logistic regression model adjusts coefficients β_j for each gene j according to the function:

$$P(y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{j=1}^{p} \beta_j x_j)}}$$

The importance of a gene j is given by the absolute value of its coefficient:

$$Importance_j = |\beta_j|$$

where β_j is the weight associated with gene j. A larger coefficient in absolute value indicates a more significant impact on prediction.

Support Vector Machine (SVM):
For linear kernel SVM, the model adjusts a weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_p)$, with a decision function:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

The importance of gene j is given by:

$$Importance_j = |w_j|$$

where w_j is the weight associated with gene j. Genes with high weights in absolute value are considered the most discriminative.

Decision Tree:
In a decision tree, gene importance is calculated according to the impurity reduction (e.g., Gini index) contributed by each gene during tree splits. For a gene j, importance is defined as:

$$Importance_j = \sum_{t \in Nodes(j)} \frac{N_t}{N} \cdot \Delta I_t$$

where:

- t iterates through all nodes where gene j is used,
- N_t is the number of samples at node t,
- N is the total number of samples,
- ΔI_t is the impurity reduction achieved at node t.

Random Forest:
Random forest aggregates gene importance across multiple trees. The importance of gene j is the average of the importances calculated across all trees:

$$Importance_j = \frac{1}{T} \sum_{t=1}^{T} Importance_{j,t}$$

where T is the number of trees and $Importance_{j,t}$ is the importance of gene j in tree t, calculated in the same manner as for a decision tree.

k-Nearest Neighbors (k-NN):
The k-NN model is non-parametric and does not directly provide coefficients for each gene. To estimate gene importance, we use *permutation importance*:

$$Importance_j = E[Score_{original} - Score_{permuted}(j)]$$

where:

- $Score_{original}$ is the model performance on the original data,
- $Score_{permuted}(j)$ is the performance after randomly permuting the values of gene j.

This method measures how much permuting a gene degrades performance, thus indicating its importance.

Permutation Importance (General):
Permutation importance is also applied to SVM, decision tree, and random forest to validate the obtained importance. The formula is the same as above:

$$Importance_j = E[Score_{original} - Score_{permuted}(j)]$$

It is robust and applicable to any model, even non-linear or non-parametric ones.

Results and Discussion

Experimental Process

Dataset Preparation

We obtained the *Gene Expression of Colon Cancer* dataset from Kaggle, containing 804 tissue samples with expression values for 60 genes. Our protocol followed these steps:

1. Loaded the dataset using `pandas` and examined its structure
2. Verified data completeness (0 missing values detected)
3. Encoded the target variable `tissue_status` ("normal" = 0, "tumoral" = 1)
4. Applied z-score standardization to normalize gene expression values:

$$z = \frac{x - \mu}{\sigma} \tag{2}$$

5. Split the dataset into training (75%) and test (25%) sets using stratified sampling

Feature Selection Protocol

We implemented an iterative feature selection procedure:

1. Calculated gene importance using five different models:
   - Logistic Regression: Extracted absolute coefficient values $|\beta_j|$
   - Linear SVM: Used weight magnitudes $|w_j|$
   - Decision Tree: Measured impurity reduction
   - Random Forest: Aggregated tree-based importance scores
   - KNN: Applied permutation importance
2. Ranked genes by average importance across all five models
3. Eliminated highly correlated genes (correlation coefficient ≥ 0.70)
4. Started with top 5 most important genes
5. Assessed model performance via 5-fold cross-validation
6. Incrementally added next highest-ranked gene until performance equaled full gene set

Model Training and Evaluation

The experimental evaluation consisted of:

1. Training five classifier types on the selected gene subset:
   - Logistic Regression with L2 regularization
   - Support Vector Machine with RBF kernel
   - Decision Tree with optimized depth
   - Random Forest with 100 estimators
   - k-Nearest Neighbors with k=5
2. Optimizing hyperparameters via grid search with 5-fold cross-validation
3. Computing performance metrics on the test set:
   - Recall and precision for both tissue classes
   - F1-score for balanced assessment
   - Area Under ROC Curve (AUC)
   - Confusion matrices
4. Constructing cumulative confusion matrices through k-fold validation
5. Selecting final model based on highest F1-score with minimal gene count

Computational Environment

The training and evaluation of the classification models were performed on **Google Colab** using the **Google Compute Engine backend**. The environment included a **Python 3** runtime, **12.7 GB of RAM**, and approximately **107 GB of disk space**. A **Tesla T4 GPU** was used to accelerate the training process. Figure **??** illustrates the Colab resource panel during experimentation.

Reduced Datasets for Classification Models

New Datasets The reduced datasets were obtained after removing redundant genes (correlation > 0.70) and selecting the most important genes for each model.

Table 2: Reduced Datasets by Model

Model	Retained Genes	Number	Removed Genes
k-NN	SLC7A5, RNF43, DAO, NEURL1B	4	{UGP2}
Logistic Regression	SLC7A5, RNF43, DAO, FOXF2, DDIT4	5	{UGP2}
SVM	SLC7A5, RNF43, DDIT4, DAO, FOXF2	5	{UGP2}
Decision Tree	RNF43, UGP2, NCAPH, DHRS11	4	{set()}
Random Forest	RNF43, DAO, UGP2	3	{SLC7A5, NEURL1B}

Comparative Analysis

- **Number of genes**: Random Forest (3 genes) is the most economical, followed by k-NN and Decision Tree (4 genes), then Logistic Regression and SVM (5 genes).
- **Common genes**: RNF43 and UGP2 appear in all models; DAO appears in 4/5 models.
- **Differences**: Decision Tree retains UGP2 and adds NCAPH/DHRS11; SVM includes DDIT4/FOXF2; Random Forest excludes SLC7A5/NEURL1B.
- **Interpretation**: Random Forest optimizes with fewer genes due to ensembling. k-NN, Logistic Regression, and SVM require more features. Decision Tree is more structure-dependent.

Table 3: Gene Frequency in Models

Gene	Number of Occurrences
RNF43	5
UGP2	5
DAO	4
SLC7A5	3
NEURL1B	2
DDIT4	2
FOXF2	2
NCAPH	1
DHRS11	1

Model Performance Analysis

Without Noise (Table 4)

When noise is removed:

- All models (Logistic Regression, KNN, SVM, Decision Tree, and Random Forest) achieve perfect scores (1.0) across all metrics.
- This confirms that dataset noise was the main factor influencing performance in the noisy setting.

Table 4: Model Performance Metrics without Noise

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	1.0	1.0	1.0	1.0
KNN	1.0	1.0	1.0	1.0
SVM	1.0	1.0	1.0	1.0
Decision Tree	1.0	1.0	1.0	1.0
Random Forest	1.0	1.0	1.0	1.0

With Noise (Table 5)

With noise included:

- Logistic Regression, SVM, and Random Forest maintain perfect performance (1.0).
- KNN and Decision Tree slightly decline, showing 0.995 in Accuracy and 0.99 in Recall, while Precision remains perfect.

Table 5: Model Performance Metrics with Noise

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	1.000	1.0	1.00	1.000
KNN	0.995	1.0	0.99	0.994
SVM	1.000	1.0	1.00	1.000
Decision Tree	0.995	1.0	0.99	0.994
Random Forest	1.000	1.0	1.00	1.000

Feature Efficiency Analysis

Based on the reduced datasets information:

- **Random Forest** uses the fewest features (3 genes: RNF43, DAO, UGP2)
- **KNN** and **Decision Tree** use 4 genes each
- **Logistic Regression** and **SVM** require 5 genes each

Results

Random Forest emerges as the optimal model for this classification task because:

- It achieves perfect performance metrics even in the presence of noise
- It requires the fewest features (only 3 genes)
- Its ensemble approach provides greater robustness against noise

Confusion Matrices and Performance Metrics

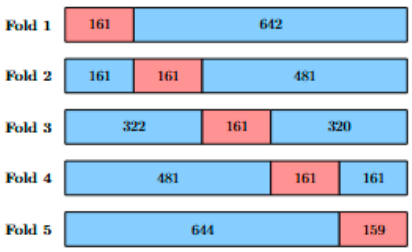

Figure 4: 5-Fold Cross-Validation for 803 Observations.

The following confusion matrices summarize the classification results for each model: Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, and Support Vector Machine (SVM).

Table 6: Confusion Matrices of Different Models

Model	TruePredicted	**0**	**1**
Logistic Regression	0	400	2
	1	4	398
KNN	0	401	1
	1	5	397
Decision Tree	0	399	3
	1	2	400
Random Forest	0	400	2
	1	2	400
SVM	0	401	1
	1	3	399

The main performance metrics derived from the confusion matrices are summarized in the following table, allowing a global comparison of the classification performance of each model.

Table 7: Comparison of Performance Metrics

Model	Precision	Recall	F1-Score	Accuracy (%)
Logistic Regression	0.9950	0.9900	0.9925	99.25
KNN	0.9975	0.9876	0.9925	99.25
Decision Tree	0.9926	0.9950	0.9938	99.38
Random Forest	0.9950	0.9950	0.9950	99.50
SVM	0.9975	0.9925	0.9950	99.50

Discussion of Results

Our **gene selection approach** demonstrated remarkable efficacy in distinguishing between **normal and tumor tissues** based on **gene expression profiles**. By implementing **feature selection techniques**, reducing **redundancy** through a **0.70 correlation threshold**, and employing rigorous **5-fold cross-validation**, we maintained classification performance comparable to the full **60-gene set** while drastically reducing the **feature space**.

The **Random Forest classifier** exhibited superior performance utilizing just **three genes** (**RNF43**, **DAO**, **UGP2**), showcasing the strength of **ensemble methods** in capturing complex biological relationships. In contrast, **Logistic Regression** and **SVM** required **five genes** to achieve similar results, reflecting their differential sensitivity to data characteristics and feature interdependencies.

Notably, certain genes emerged as **consistent predictors** across multiple model configurations. **RNF43** appeared universally across all optimal models, strongly suggesting its fundamental role in **colon cancer pathophysiology**. Similarly, **DAO** and **UGP2** demonstrated high frequency in selected feature subsets, indicating their potential significance as **diagnostic biomarkers**.

For addressing **class imbalance**, **random oversampling** proved effective despite its simplicity compared to more sophisticated methods like **SMOTE**. The **confusion matrices** derived from cross-validation confirmed that this straightforward rebalancing approach enhanced class representation without compromising model integrity, particularly advantageous given our moderate **dataset size of 803 samples**.

Comprehensive evaluation through multiple metrics—**recall**, **precision**, **F-score**, and **AUC**—corroborated the **robustness** of our final model. The consistent performance across validation iterations and model architectures underscores the **reliability** of our analytical pipeline, suggesting its applicability to analogous datasets in **bioinformatics** and **personalized medicine** contexts. These findings highlight the potential for **minimal gene signatures** to facilitate efficient and accurate **cancer diagnostics**.

Conclusion

In this study, we successfully applied various **machine learning** classifiers to gene expression data for the purpose of distinguishing between **normal** and **tumoral colon**

tissues. Our approach combined efficient **feature selection**, effective handling of **class imbalance**, and robust model validation strategies. The results demonstrated that a highly accurate classification could be achieved using a **reduced set of genes**, particularly through the use of the **Random Forest classifier**. Notably, **RNF43**, **DAO**, and **UGP2** emerged as consistently relevant features, indicating their potential value in **biomedical research** and **clinical applications**.

Furthermore, this study reinforces the importance of **dimensionality reduction** and the advantages of using **ensemble methods** in high-dimensional biomedical datasets. The success of this analytical pipeline suggests its generalizability to other **bioinformatics** problems and offers promising directions for future research in **personalized medicine** and **cancer biomarker discovery**.

References

[1] Kaggle, *Gene Expression of Colon Cancer Dataset*. Available at: https://www.kaggle.com/datasets/ambujtripathi/genetic-expression-of-colon-cancer

[2] Hnioua, A., *Gene Expression Analysis in Colorectal Cancer Using Machine Learning*. GitHub repository: https://github.com/hnioua/Gene-Expression-Analysis-in-Colorectal-Cancer-Using-Machine-Learning

[3] Stack Overflow, *Community Forum for Programming and Data Science*. Available at: https://stackoverflow.com

[4] Cross Validated (StackExchange), *Q&A for People Interested in Statistics, Machine Learning, Data Analysis*. Available at: https://stats.stackexchange.com

[5] Scikit-learn Documentation, *Machine Learning in Python*. Available at: https://scikit-learn.org/stable/documentation.html

[6] Towards Data Science, *Articles and Tutorials on Machine Learning*. Available at: https://towardsdatascience.com

[7] Overleaf – LaTeX Math Formula Guide. Available at: https://www.overleaf.com/learn/latex/Mathematical_expressions