Faculty of Sciences

Faculty of Mathematics, Statistics, and Computer Science

Statistical Machine Learning — Project

Due Date: Sat, 31 Jan 2026 (23:59)

Introduction.

The aim of this project is to introduce students to high-dimensional data modeling, specifically with genomic data. One application is the breast cancer dataset from METABRIC, which we will use. Most of us know someone who struggled with breast cancer, or at least heard about the struggles facing patients who are fighting against breast cancer. Breast cancer is the most frequent cancer among women, impacting 2.1 million women each year. Breast cancer causes the greatest number of cancer-related deaths among women. In 2018 alone, it is estimated that 627,000 women died from breast cancer. The most important part of a process of clinical decision-making in patients with cancers, in general, is the accurate estimation of prognosis and survival duration. Breast cancer patients with the same stage of the disease and the same clinical characteristics can have different treatment responses and overall survival, but why? Cancers are associated with genetic abnormalities. Gene expression measures the level of gene activity in a tissue and gives information about its complex activities. Comparing the genes expressed in normal and diseased tissue can bring better insights into the cancer prognosis and outcomes. Using machine learning techniques on genetic data has the potentials of giving the correct estimation of survival time and can prevent unnecessary surgical and treatment procedures. In this project, we can assess various things such as model accuracy, calibration, validation, explainability, and feature importance of covariates. In this project, you will:

- Choose a prediction task: classification for death from cancer (yes/no) or regression for tumor size.
- Explore and preprocess high-dimensional gene expression data.
- Apply baseline regularization methods such as Lasso and Ridge for prediction.

- Implement an advanced high-dimensional statistical algorithm unique to your group (e.g., Adaptive Lasso, SCAD penalty, Group Lasso) and explain the algorithm and mathematics.
- Perform feature selection and evaluate model performance (including new metrics like AUC).
- Interpret results using SHAP for model explainability and identify important covariates.
- Compare methods and analyze their strengths and limitations on real-world genomic data.

Part 0. Background & Dataset.
The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database is a Canada-UK Project which contains targeted sequencing data of 1,980 primary breast cancer samples. Clinical and genomic data was downloaded from cBioPortal. The dataset was collected by Professor Carlos Caldas from Cambridge Research Institute and Professor Sam Aparicio from the British Columbia Cancer Centre in Canada and published on Nature Communications (Pereira et al., 2016). It was also featured in multiple papers including Nature and others. The dataset is publicly available at:

- Dataset link

Important note on data access.
You may download the dataset manually to your local system if desired. However, it is **strongly recommended** to use the Kaggle-hosted version of the dataset together with the **Kaggle API**, especially when working in **Google Colab** or directly on **Kaggle**. This approach allows you to download and load the data programmatically with minimal effort, improves reproducibility, and avoids unnecessary manual downloading and file management. Our task is to predict whether the patient's death was due to cancer (death_from_cancer: yes/no) using gene expression profiles and clinical features. Optionally, for a more complex task, you may extend this to regression for predicting tumor size or survival time.

Part 1. Data Exploration and Feature Selection.
High-dimensional genomic data requires careful preprocessing and feature selection to handle the curse of dimensionality.

- Load the dataset.

- Perform exploratory data analysis: summarize statistics, handle missing values, visualize distributions of key variables, visualize the distribution of clinical columns in the dataframe, and also visualize these features by the `overall_survival` label (whether the patient is alive or dead) using different colors for each class.

- Do correlation analysis and see if you have highly correlated features (above some threshold), report them, and discuss what issues these can cause.

- Apply PCA to 2–3 dimensions to see if there is any pattern in the data, coloring the points with the target `overall_survival` label.

- Scale the data and report if you remove any feature due to any reason.

- Report the initial and final feature dimensionality, and discuss selected features.

Part 2. Baseline Classification Models: Lasso and Ridge Logistic Regression.

- Split the dataset into training and test sets with stratification, ensuring that the class proportions of the target variable are preserved across splits. Perform 5-fold cross-validation on the training data to tune hyperparameters, selecting the optimal values based on the mean (or median) cross-validated performance, and evaluate the final model once on the held-out test set.

- Train Lasso and Ridge logistic regression models, tuning hyperparameters such as the regularization strength (alpha).

- Evaluate performance using appropriate classification metrics, including precision, recall, F1 score, accuracy, and ROC-AUC. Report both the mean and standard deviation of these metrics across the cross-validation folds.

- Select the optimal hyperparameters based on cross-validation, retrain each model on the full training data using the selected hyperparameters, analyze the selected features for the Lasso model, and report the best-performing baseline classification model.

- Repeat the above procedure using three feature subsets: (i) non-genetic features only, (ii) genetic features only, and (iii) the combined set of genetic and non-genetic features. Compare the performance of the best-performing model across these three feature subsets and report the best model of each family along with its corresponding feature subset.

Part 3. Ensemble and Nonlinear Classification Models

- Implement and evaluate the following classification algorithms which some of them suitable for high-dimensional data: K-Nearest Neighbors (KNN), Support Vector Machines (SVM) with an appropriate kernel for the problem, Decision Trees, Random Forests, and AdaBoost.

- For Decision Trees and Random Forests, tune only the maximum tree depth (`max_depth`). For AdaBoost, use KNN and Decision Trees as weak learners and tune the number of estimators (`n_estimators`) and learning rate (`learning_rate`). For SVM, tune the regularization parameter (`C`) and, if applicable, the kernel parameters; for KNN, tune the number of neighbors (`k`).

- Follow the same procedure as in Part 2: stratified train/test splits, 5-fold cross-validation for hyperparameter tuning based on mean (or median) performance, retraining the final model on the full training data, and evaluation on the held-out test set using the same classification metrics. Report both mean and standard deviation of metrics across CV folds.

- Repeat the procedure for three feature subsets: (i) non-genetic features only, (ii) genetic features only, and (iii) the combined set of genetic and non-genetic features.

- For each algorithm family, report the best-performing model along with its corresponding feature subset, and compare performance across models and feature subsets.

Part 4. Model Evaluation, Comparison, and Interpretation with SHAP.

- Compare the best-performing model from each model family trained in the previous parts, reporting the model itself along with the corresponding feature subset used (non-genetic, genetic, or combined), and present its performance metrics (mean and standard deviation across cross-validation folds as well as test set results).

- SHAP helps to interpret complex models by quantifying the contribution of each feature to the predictions. You can find more details and examples of its usage in R at the following link: `shapr`.

- For the overall best model selected, compute SHAP values for the most important covariates, visualize the results (e.g., summary plots, dependence plots), and discuss the top contributing features.

Important Notes:

1. At the beginning of your code, use `set.seed(123)` to ensure reproducibility of results.
2. When comparing models, if there is a significant challenge; use accuracy as your baseline metric for reference.
3. Submit your code as an `Rmd` file and include the data so that it can be run directly.
4. The project can be completed individually or in groups of two. If you choose to work in a group, provide the group members' names in this form: Google Form by January 10 (20 Dey).