Detecting Breast Cancer Type

The problem I'm trying to solve:

A gene expression data is given labeled with one of the classes of Breast Cancer. Given the Gene expression data, the algorithms I use will predict the cancer type. I am training three kinds of models and evaluating each model's accuracy of prediction.

How does the SVC algorithm work?

The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

The objective of a Linear SVC (Support Vector Classifier) is to fit the data you provide, returning a "best fit" hyperplane that divides or categorizes your data. From there, after getting the hyperplane, you can then feed some features to your classifier to see what the "predicted" class is.

How does the MultinomialLogisiticRegression algorithm work?

An extension of logistic regression that adds native support for multi-class classification problems. It is limited to two-class classification problems. Some extensions like one-vs-rest can allow logistic regression to be used for multi-class classification problems, although they require that the classification problem first be transformed into multiple binary classification problems.

Instead, the multinomial logistic regression algorithm involves changing the loss function to cross-entropy loss and predicting probability distribution to a multinomial probability distribution to natively support multi-class classification problems.

How does the Decision Trees Algorithm work?

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node

Briefly describe what you did to tune to each model and how the hyperparameters you tuned impact that model.

Parameters that define the model architecture are referred to as hyperparameters and the process of searching for the ideal model architecture is referred to as *hyperparameter tuning*.

Grid search is arguably the most basic hyperparameter tuning method. With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results

For SVC model tuning, I used Kernel: Specifies the kernel type to be used in the algorithm, gamma: Kernel coefficient, and C: regularization parameter.

Regularization generally refers to the concept that there should be a complexity penalty for more extreme parameters.

For MultinomialLogisticRegression, I used tol: tolerance stopping criteria, C: regularization parameter.
For decision tree model tuning, I used two hyperparameters max_depth: What should be the maximum allowable depth for each decision tree and max_features: The number of features to consider when looking for the best split