**Part 1. Normal and Tumor Match Pair Analysis**

1. Using SciKit Learn build a machine learning classifier that takes RNAseq profiles from matched normal tumor pairs and classifies the sample as Normal or Tumor. Compare the **nt.coding.csv** vs the **nt.all.csv**.
2. Using model selection methods of your choice determine which classical ML method performs best on the NT classification problem.

To tackle these two parts of the problem, I initially built a machine learning pipeline (in the ml\_functions.py file) that instantiates several of the most common classical machine learning models (i.e. Random Forest, Logistic Regression, Naïve Bayes, and Decision Trees). The pipeline includes grids for different hyper-parameter combinations to see each model with (each tested in a loop function called run\_simple\_loop). From there, I built a function called clf\_loop (also called by run\_simple\_loop) that builds out the 5-fold cross validation – along with detailed evaluation metrics (auc-roc, precision at thresholds: [1, 2,5,10,20,30,50], recall at thresholds: [1, 2,5,10,20,30,50], F1 at thresholds: [1, 2,5,10,20,30,50], and confusion matrix) across each of the folds. Then, for each unique model/parameter combination, I averaged the evaluation metrics – with particular focus on auc-roc score – and compared the results.

For nt.coding, the best performing machine learning model that I tested was a Random Forest classifier with n\_estimators: 200, max\_depth: 5, max\_features: sqrt, min\_samples\_split: 2, n\_jobs = -1). This had an auc-roc score of .9906285.

For nt.all, the best performing machine learning model that I tested was also a Random Forest classifier with n\_estimators: 200, max\_depth: 10, max\_features: sqrt, min\_samples\_split: 2, n\_jobs = -1). This had an auc-roc score of .99037467.

1. Using feature selection methods of your choice determine a < 100 gene signature that can be used to classify Normal vs Tumor.

I used sklearn’s SelectKBest method, using a chi-squared test evaluator, the find the 99 most relevant features used to classify normal vs. tumor. I ran this feature selection algorithm for each of the datasets – the corresponding outputs (including the list of 99 most relevant features for each dataset) are in the attached Python notebook.

I then ran the previous constructed machine learning pipeline on the 99 gene signature separately for nt.coding vs. nt.all and compared their corresponding evaluation metrics. The best performing model, according to average auc-roc score after 5-fold cross validation, achieved a score of .9834 in nt.coding while the best performing model for .98327 in nt.all. These results are quite both quite strong – and are only slightly less well-performing than the results achieved when running the whole datasets (with all features included) on the nt.coding and nt.all datasets.

1. Using **Keras**, build a deep learning classifier that performs the same classification task, and determine the learning curve (relationship of number of training samples to prediction accuracy) for your network, recommend using at least 10 training set sizes to estimate the learning curve.
2. Extra Credit: Use the TPOT autoML system to search for a better solution to part a.

I u

Part 2. **Cancer Type Classifier for 18 Common Tumor Types.**

1. Using SciKit Learn build a machine learning classifier that classifies Cancer Type from the **type.coding.csv** and **type.all.csv** files. Compare the coding vs all genes cases.
2. Using model selection methods of your choice, determine which classical ML method performs best.
3. Using feature selection methods of your choice, determine a < 100 gene signature that can be used to classify tumor type.
4. Using **Keras**, build a deep learning classifier that performs the same classification task, and determine the learning curve (relationship of number of training samples to prediction accuracy) for your network.