PCA and Under-sampling

Program 2 is working with the same data as Program 1. We will be assessing the impact of applying PCA to it as well as looking at the issue of class imbalance.

Once again you will use the attached csv file that has gene expression data for 249 tissue samples. Each is labeled with one of the following classes (https://www.breastcancer.org/symptoms/types/molecular-subtypes):

0 – Luminal B
1 – Luminal A
2 – Normal-like
3 – Basal-like
4 – HER2+

As in Program 1, you will create and use a Jupyter notebook to complete this assignment.

Part 1:

All of your evaluations will use the Random Forest algorithm and will following the basic evaluation structure described below:

1. You will evaluate the performance of Random Forests on this data under the following condition.
    a. Using the full data set
    b. Using PCA covering 95% of the variance
    c. Using PCA with 25, 16, 9, and 4 principal components.
2. You will print (using Python) a table that clearly shows the user the average and standard deviation of the overall accuracy for each of the situations above. The performance statistics will be computed using 10 runs for each of the six data scenarios.
3. Each run will consist of the following:
    a. Create a stratified 80/20 train/test split.
    b. Perform any needed feature transformation on the training data.
    c. Using the training data, tune (gridSearchCV) the following Random Forest parameters: n_estimators and criterion. Be sure to include the default values as part of the process.
    d. Using the best model resulting from the tuning process, train a Random Forest using the training data.
    e. Perform any needed feature transformation on the test data.
    f. Evaluate the trained model using the test data and store accuracy for later use.
4. You will need to understand and appropriately use both pca.fit_transform and pca.transform.

Part 2:

Part 2 will focus on predicting a single class, HER2+. You will generate a vector of data labels that is appropriate for binary classification. Consider label 4 to be a positive example and the other labels as negative examples. The issue focused on in Part 2 is class imbalance.

1. You will evaluate the performance of Random Forests on this data under the following condition.
    a. Using the full data set with no adjustments for class imbalance
    b. Using features available in the Random Forest algorithm to address class imbalance
        i. class_weight="balanced"
        ii. class_weight="balanced_subsample"
    c. Using other sampling approaches to address class imbalance. Note that over/under-sampling is only applied to the training data.
        i. 50% random under-sampling of the majority class
        ii. 100% over-sampling of the minority class using SMOTE
        iii. Combining both of the above.
2. You will print (using Python) a table that clearly shows the user the average and standard deviation of the precision and recall for each of the situations above. The performance statistics will be computed using 10 runs for each of the six data scenarios.
3. Each run will consist of the following:
    a. Create a stratified 80/20 train/test split.
    b. Perform any needed data transformation on the training data.
    c. Using the training data, tune (gridSearchCV) the following Random Forest parameters: n_estimators and criterion. Be sure to include the default values as part of the process.
    d. Using the best model resulting from the tuning process, train a Random Forest using the training data.
    e. Evaluate the trained model using the test data and store precision and recall for later use.
4. SMOTE information/example:
    a. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
    b. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/
5. Finish Part 2 by adding a text cell that describes the following (in your own words):
    a. What problem(s) can be caused by class imbalance and why?
    b. How each of the five approaches used attempts to mitigate problems with class imbalance.