Classification of malignant/bening breast tissue

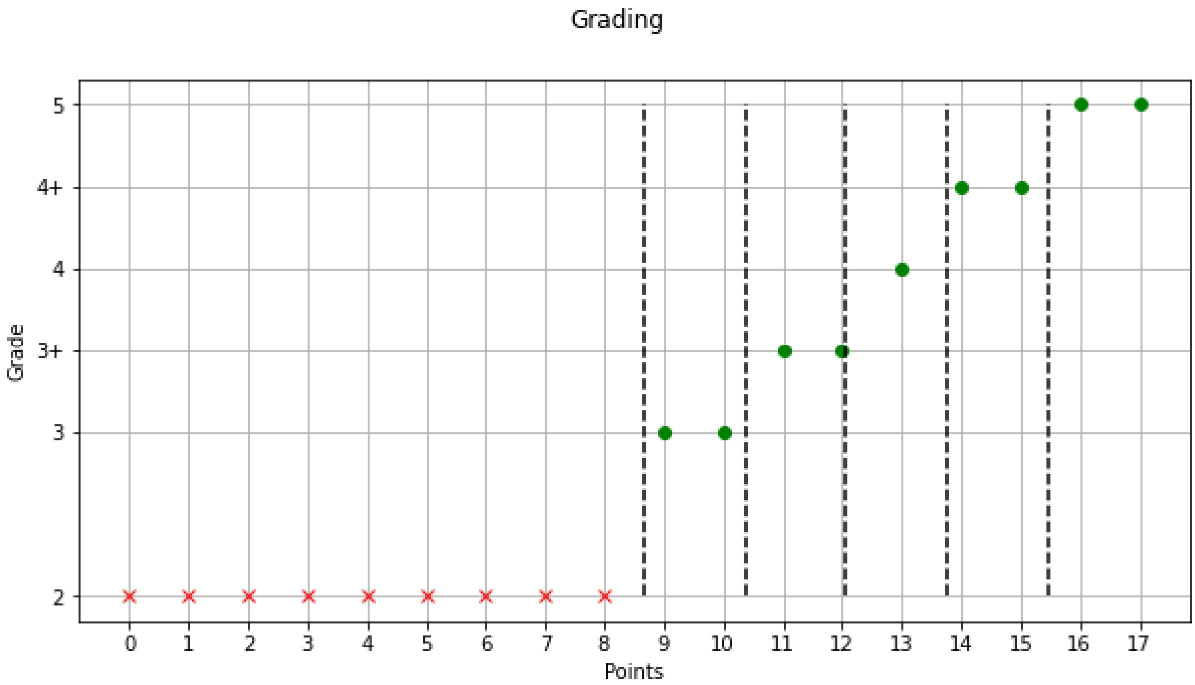
The goal is to setup a machine learning task according to the outline below. The data are composed of 569 examples, out of which 357 is from bening cells and 212 from malignant (cancerous) cells. Each example (cell) is described by 30 features extracted from a histological image taken from the breast, in the order:

'mean radius', 'mean texture', 'mean perimeter', 'mean area', 'mean smoothness', 'mean compactness', 'mean concavity', 'mean concave points', 'mean symmetry', 'mean fractal dimension', 'radius error', 'texture error', 'perimeter error', 'area error', 'smoothness error', 'compactness error', 'concavity error', 'concave points error', 'symmetry error', 'fractal dimension error', 'worst radius', 'worst texture', 'worst perimeter', 'worst area', 'worst smoothness', 'worst compactness', 'worst concavity', 'worst concave points', 'worst symmetry', 'worst fractal dimension'.

The task involves data loading, dimensionality reduction, train/test data split, classifier training, calculation of classification metrics, display of results.

1. Load the data. Feature values are in data.npy; targets in targets.npy. Auxilliary data that can be used (but are not necessary strictly) are feature_names.npy and class_names.py. **(1p)**
2. Display a scatter plot for all the samples and the features: 'mean area' on the X i 'worst fractal dimension' on the Y axis. Add axes labels and plot title. **(2p)**
3. Use the principal component analysis on the dataset. Then, reduce it's dimensionality to the two most important features according to PCA. Further steps should be performed for the reduced dataset. **(3p)**
4. Split the data into train/test sets in a 80:20 ratio using a function to do it automatically and randomly. Set the random_state parameter according to the teacher's recommendation. **(1p)**
5. Train a support vector machine classifier. Set the regularization constant C=0.5 and use a nonliner kernel (e.g. Radial Basis Functions). Then classify the test examples using the trained SVM. **(3p)**
6. Compute the confusion matrix for the test set. Display the result in the form of a message: "There were X true negatives, X true positives, X false negatives and X false positives in the test set results". Substitute X's with the correct number from the confusion matrix. **(3p)**
7. Display the reduced training and test sets using the targets' vector to colorize the examples in the plot. Use a different colormap for the training and test examples. Label the axes correctly (1st principal component, 2nd principal component), add title to the plot: „Breast cancer classification". Display the decision boundary and margins. **(4p)**
8. *Extra task: Use k-means for clustering the training samples instead of the SVM in point 5 (1p). Then, predict the cluster for the test samples (1p). Interpreting the first cluster as bening and the second as malignant, compute the confusion matrix with respect to the targets' vector and display a message as in point 6. Try the same again, interpreting the cluster class reversely (first – malignant, second – beningn) (1p).*

You can get 17 points in total and grades are given according to the following points vs grade graph:



The extra task can only be performed if the regular task was finished. Points earned for the extra task (max. 3) will be added to regular points and can enhance the grade, provided that the regular grade was positive.