Bayesian Classification (**scored task: 3 points**)

Tasks (Labs 1-2)

1. Implementation (from scratch) of LDA, QDA and NB (Naive Bayes) methods for binary classification (classes 0 and 1):

   - Create three classes in Python: LDA, QDA, NB
   - Each class should have three methods:
     (a) Method `fit(X,y)`, where `X` is a design matrix, containing in each row the values of features for a single observation and `y` is a vector containing values of binary target variable for observations. This method fits the model using training data. In the case of LDA, QDA and NB training the model includes estimation of the parameters (vectors of means, covariance matrices, prior probabilities).
     (b) Method `predict_proba(Xtest)` which computes predicted posterior probabilities for class 1 for observations whose feature values are in rows of the matrix `Xtest`.
     (c) Method `predict(Xtest)` which assigns the predicted class (0 or 1) for observations whose feature values are in rows of the matrix `Xtest`.
     (d) Method `get_params` that returns a list containing the estimated parameters.

2. Comparison of LDA, QDA and NB methods on simulated data.

   - There are two parameters which will vary in experiments: a and ρ. Generate training and testing data in the following way.
     - **Scheme 1:** Each dataset contain $n = 1000$ observations, $p = 2$ features and a binary variable that is generated from the Bernoulli distribution with probability of success 0.5. Features of the observations from the class 0 are generated independently from a normal standard distribution (mean 0, variance 1). Features of the observations from the class 1 are generated independently from a normal distribution (mean a, variance 1). **LDA assumptions are satisfied.**
     - **Scheme 2:** Each dataset contain $n = 1000$ observations, $p = 2$ features and a binary variable that is generated from the Bernoulli distribution with probability of success 0.5. Features of the observations from the class 0 are generated from a two-dimensional normal distribution (mean 0, variance 1, correlation ρ). Features of the observations from the class 1 are generated from a two-dimensional normal distribution (mean a, variance 1, correlation $-\rho$). **LDA assumptions are not satisfied.**
   - Compare LDA, QDA, and NB for both schemes (compute accuracy on the testing set) for fixed value $\rho = 0.5$ and different values of $a = 0.1, 0.5, 1, 2, 3, 5$. Repeat the experiment for different train/test splits and generate boxplots showing the values of accuracy for each method and each value of the parameter a. Save the results in the file `BayesianSimulatedData1.pdf`

- Compare LDA, QDA, and NB for both schemes (compute accuracy on the testing set) for fixed value $a = 2$ and different values of $\rho = 0, 0.1, 0.3, 0.5, 0.7, 0.9$. Repeat the experiment for different train/test splits and generate boxplots showing the values of accuracy for each method and each value of the parameter ρ. Save the results in the file `BayesianSimulatedData2.pdf`
- For one chosen setting of parameters (e.g. $a = 2$, $\rho = 0.5$) generate a scatter plot showing observations from training set. Mark observations belonging to different classes using two different colors and two different symbols. Draw curves that separate classes for LDA and QDA. Save the results in the file `BayesianSimulatedData3.pdf`

3. Comparison of LDA, QDA and NB methods on real data.

   - Choose 3 datasets which are available on e.g., UCI repository `https://archive.ics.uci.edu/ml/index.php` or OpenML repository `https://www.openml.org/` related to binary classification problem. Please only focus on datasets with numerical features.
   - Compare LDA, QDA, and NB. Split data into training set and test set. Train the model on the train set and compute accuracy on the test set. Repeat the experiment for different train/test splits and generate boxplots showing the values of accuracy for each method. Save the results for three datasets in the file `BayesianReal.pdf`