3) Principal Component analysis (PCA)

```
Function PCA (Data, D, Integer K):
    Standardized ← STANDARDISE_DATA (D)
    CoVMatrix ← COMPUTE_COVARIANCE_MATRIX
                (Standardized)
    Eigenvalues, Eigenvectors ← Eigen - Decomposition
                (CovMatrix)
    Sorted ← Sort - Eigen Vectors - By - Eigen values (Eigenvalues,
             Eigenvectors)
    Top_K_Vectors ← Select - Top_K - Eigenvectors (Sorted, K)
    Reduced Data ← Project_Data (Standardized,
                   Top_K_vectors)
    Return Reduced Data

function STANDARDIZE_DATA (Data D):
    for each feature in D:
        Subtract mean and divide by Standard
                                    variation
    Return Standardized D

function Project_Data (Data, Eigenvectors):
    Return Matrix_Multiplication (Data, Eigenvectors)
```

4) Ada Boost (Adaptive Boosting) algorithm

```
Function ADABOOST (Dataset D, Integer T):
    Initialize weights wi = 1/n for each training
    Sample (xi, yi)
    Classifiers ← [ ]
    Alphas ← [ ]
    for t from 1 to T do:
        Classifier ht ← TRAIN-WEAK-LEARNER
                (D, weights w)
        Error εt ← Σ [wi * I(ht(xi) ≠ yi)]
        Alpha at ← 0.5 * log((1-εt)/εt)
        for each i from 1 to n do:
            wi ← wi * exp(-at * yi * ht(xi))
        normalize weights: wi ← wi / Σ wi
        Append ht to Classifiers
        Append at to Alphas
    Return Classifiers, Alphas
Function PREDICT-ADABOOST (Classifiers,
        Alphas, Instance x):
    Total ← 0
    For t from 1 to T do:
        Total ← Total + at * ht(x)
    Return SIGN(Total)
```

09.04.