```python
def fit(self, X):
    """
    entry point to the transform data to k dimensions
    standardize and compute weight matrix to transform data.
    The fit functioin returns the transformed features. k is the number of features which cumulative
    explained variance ratio meets the target_explained_variance.
    :param   m X n dimension: train samples
    :return  m X k dimension: subspace data.
    """

    self.feature_size = X.shape[1]

    # your code here
    x_std = self.standardize(X)

    mean_vect = self.compute_mean_vector(x_std)
    cov_matrix = self.compute_cov(x_std, mean_vect)
    eig_vals, eig_vects = self.compute_eigen_vector(cov_matrix)

    eig_pairs = []
    for i in range(len(eig_vals)):
        eig_pairs.append((np.abs(eig_vals[i]), eig_vects[:,i]))

    var = self.compute_explained_variance(eig_vals)
    var_sum = self.cumulative_sum(var)

    w_matrix = self.compute_weight_matrix(eig_pairs, var_sum)

    print(len(matrix_w),len(matrix_w[0]))
    return self.transform_data(X_std=x_std, matrix_w=matrix_w)
```