# **Homework 4**

AMATH 482

Christian Diangco

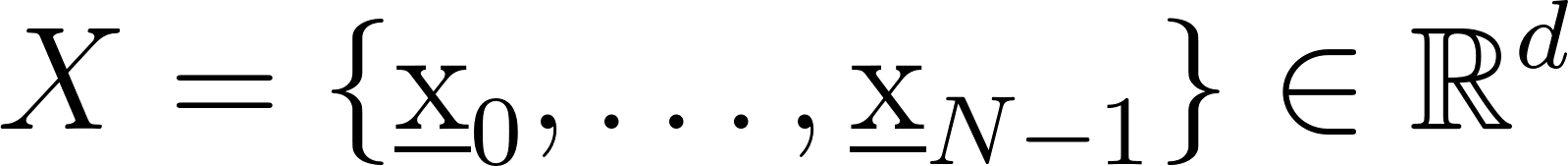
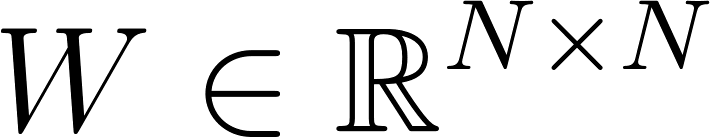
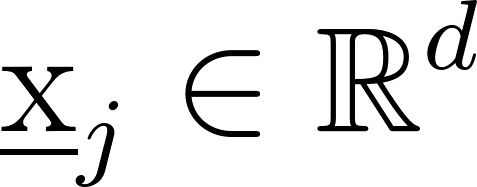
## **Abstract**

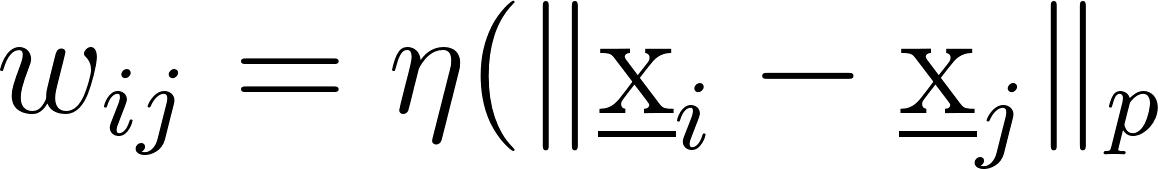
This paper covers the theory and use of spectral clustering, semi-supervised machine learning, and linear regression..The process is discussed in Sec. 3. Algorithm Implementation and Development, after which results and visualizations are displayed in Sec. 4. Computational Results.

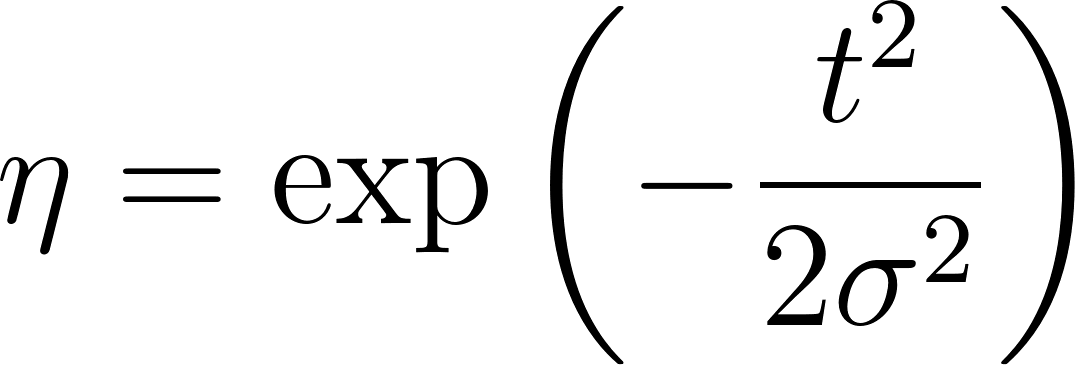
## **Sec. 1. Introduction and Overview**

In this assignment we are tasked with classifying politicians based on their voting records. To do this, we will make use of spectral clustering and semi-supervised learning techniques.

## **Sec. 2. Theoretical Background**

This assignment requires the use of spectral clustering. But first, we must cover proximity graphs and graph Laplacians. We define a dataset [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X%3D%5C%7B%5Cunderbar%7Bx%7D_0%2C%5Chdots%2C%5Cunderbar%7Bx%7D_%7BN-1%7D%5C%7D%5Cin%5Cmathbb%7BR%7D%5Ed#0) and symmetric matrix [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=W%5Cin%5Cmathbb%7BR%7D%5E%7BN%5Ctimes%20N%7D#0). Now, we define a weighted undirected graph where [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Cunderbar%7Bx%7D_j%5Cin%5Cmathbb%7BR%7D%5Ed#0) are the vertices of G and the entries [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=w_%7Bij%7D#0) of [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=W#0) denote weights that are associated to edges that connect the vertices.

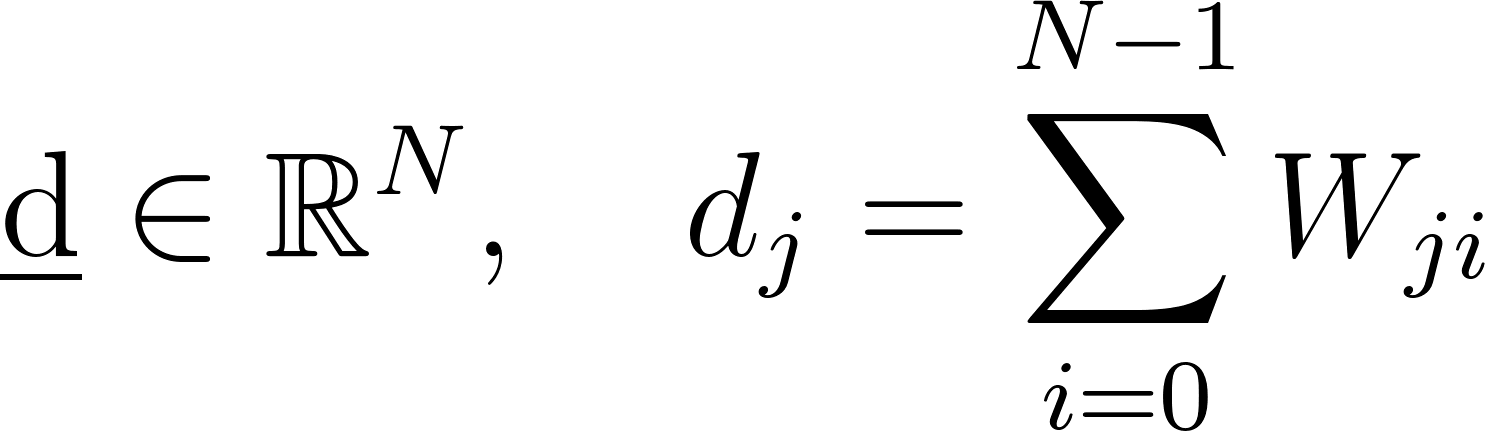
Let be denoted as the weight function. We then take [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=w_%7Bij%7D%3D%5Ceta(%5C%7C%5Cunderbar%7Bx%7D_i-%5Cunderbar%7Bx%7D_j%5C%7C_p#0) for . For this assignment,

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Ceta%3D%5Cexp%5Cleft(-%5Cfrac%7Bt%5E2%7D%7B2%5Csigma%5E2%7D%5Cright)#0).

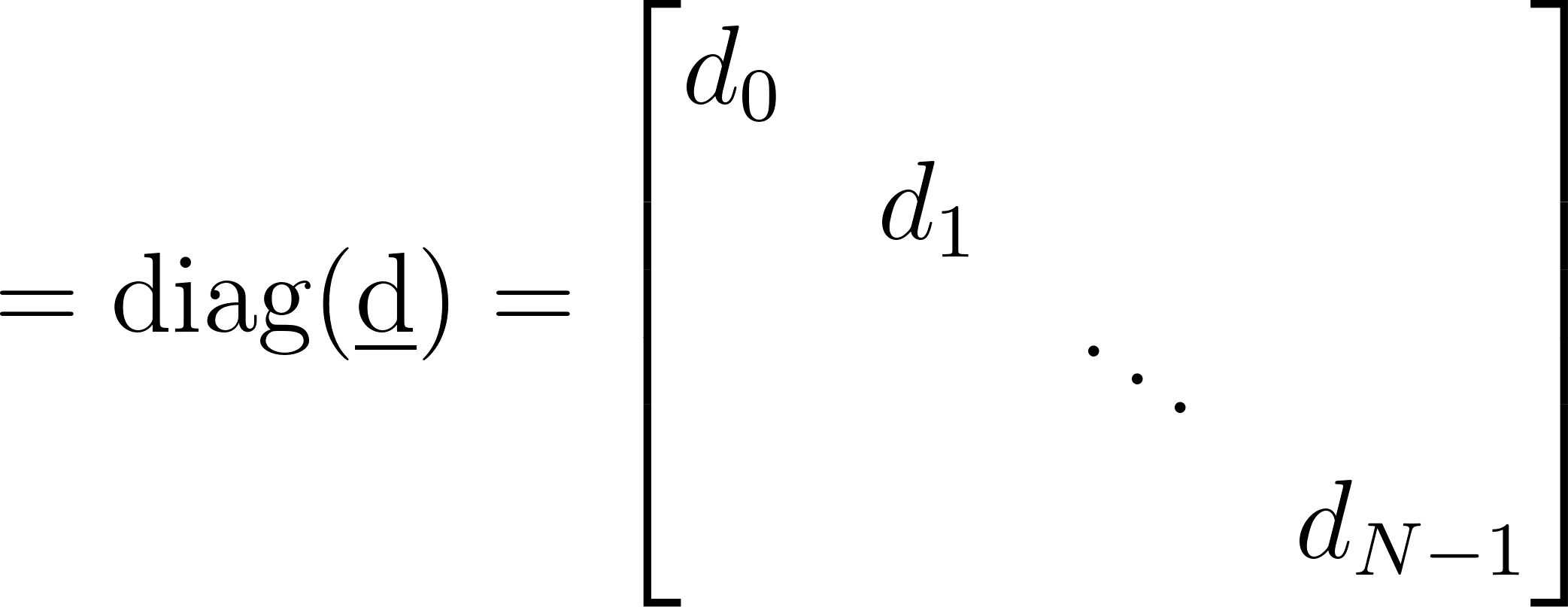
These types of graphs are known as proximity graphs.

Now that we have a matrix W, we define the graph Laplacian matrix of G.

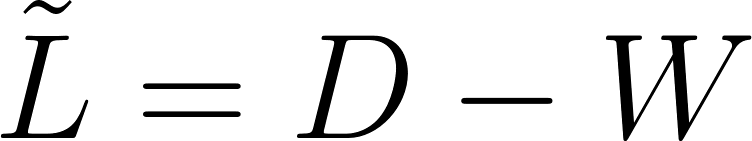
First, we define the degree vector

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Cunderbar%7Bd%7D%5Cin%5Cmathbb%7BR%7D%5EN%2C%20%5Cquad%20d_j%3D%5Csum_%7Bi%3D0%7D%5E%7BN-1%7DW_%7Bji%7D#0),

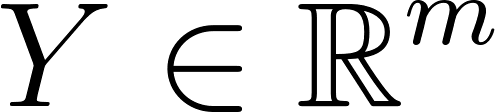
Then the diagonal degree matrix

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%3D%5Ctext%7Bdiag%7D(%5Cunderbar%7Bd%7D)%3D%5Cleft%5B%5C%5C%5C%5C%20%20%5Cbegin%7Bmatrix%7D%20d_0%5C%5C%5C%5C%20%26%20d_1%5C%5C%5C%5C%20%26%20%26%20%5Cddots%5C%5C%5C%5C%20%20%26%20%26%20%26%20d_%7BN-1%5C%5C%5C%5C%20%5Cend%7Bmatrix%7D%5Cright%5D#0),

and the unnormalized graph Laplacian

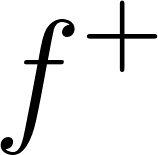
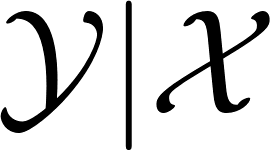
[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Ctilde%7BL%7D%3DD-W#0)**.**

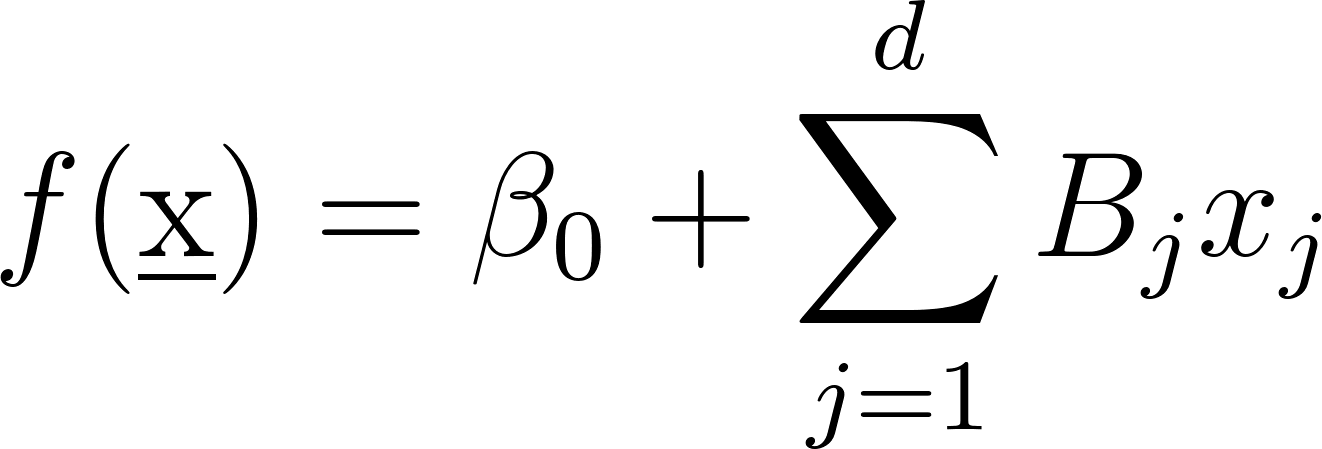
We will also utilize semi-supervised machine learning techniques, In standard machine learning terminology, we consider a set of points of features or inputs, where [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X%20%3D%20%5Cmathbb%7BR%7D%5Ed#0). We can thus again think of as a matrix [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=X%20%5Cin%20%5Cmathbb%7BR%7D%5E%7Bd%20%5Ctimes%20N%7D#0).

Associated with these features/inputs are a collection of outputs/responses . is often thought of as either a Euclidean space [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=Y%20%5Cin%20%5Cmathbb%7BR%7D%5Em#0) or space of categories e.g. , in this case the features are the wine scores. The pair of input and output sets is referred to as a training dataset. The goal of supervised learning is, given the training dataset, predict responses/output for any new input/feature (.

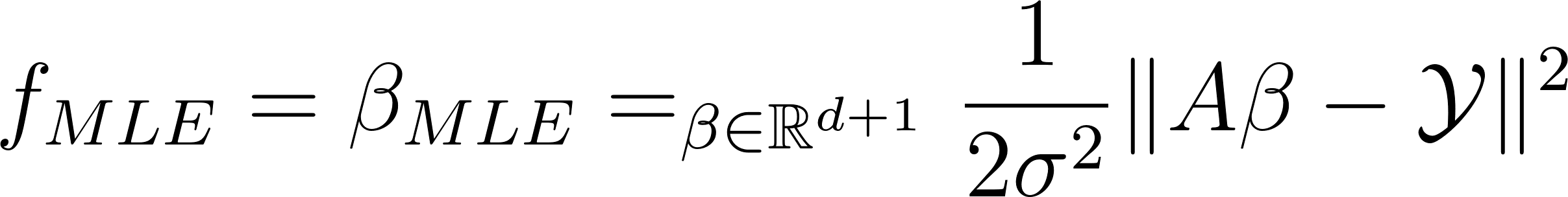
In addition, we must utilize Linear/Least Squares Regression. We assume there exists a function

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=f%5E%2B%3A%5Cmathcal%7BX%7D%5Crightarrow%5Cmathcal%7BY%7D%20%5Ctext%7B%20so%20that%20%7D%20y_j%3Df(%5Cunderbar%7Bx%7D_j)%2B%5Cvarepsilon_j#0)

Where are some noise. We can formulate an optimization problem for funding [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=f%5E%2B#0) by maximizing the likelihood of [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Cmathcal%7BY%7CX%7D#0). This is called a maximum likelihood estimator (MLE). We make some assumptions on the form of for the linear regression model,

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=f(%5Cunderbar%7Bx%7D)%3D%5Cbeta_0%2B%5Csum_%7Bj%3D1%7D%5EdB_jx_j#0)

If we assume is an affine transformation of [](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=%5Cunderbar%7Bx%7D#0), then our MLE takes the form

[](https://latex-staging.easygenerator.com/eqneditor/editor.php?latex=f_%7BMLE%7D%3D%5Cbeta_%7BMLE%7D%3D%5Cargmin_%7B%5Cbeta%5Cin%5Cmathbb%7BR%7D%5E%7Bd%2B1%7D%7D%5Cfrac%7B1%7D%7B2%5Csigma%5E2%7D%5C%7CA%5Cbeta-%5Cmathcal%7BY%7D%5C%7C%5E2#0)

MLE is a least squares solution to .

## **Sec. 3. Algorithm Implementation and Development**

After loading the data, we have to first preprocess it. I iterated through the matrix containing the original data and replaced each value with -1, +1, or 0. If a given value in the matrix is ‘republican’ or ‘n’, the value is replaced with -1. If the value is ‘democrat’ or ‘y’, the value is replaced with +1. Otherwise, the value is replaced with 0.

I then take the first column to be output y and the rest of the columns as the input features X.

To find the best value for sigma in task 2 I used 40 values in the range using NumPy’s linspace().

sigmas = np.arange(.01, 4, .01)

Next, for each of these sigma values I created the weight matrix W using the Gaussian eta() function from lecture 20.

W = eta(dist, sigma)

Here dist is the spatial distance matrix using scipy’s distance\_matrix().

dist = scipy.spatial.distance\_matrix( X, X, p =2)

The Laplacian matrix is then computed like so

d = np.sum(W, axis=1) # degree vector

D = diagonal entries of d

L = D - W

Using the second eigenvector of L we create the classifier for clustering

classifier1 = np.sign(V[:,1])

classifier2 = -np.sign(V[:,1])

I define two and use the max for the number of matches to compute the accuracy for the current sigma value.

count1 = sum(classifier1 == y)

count2 = sum(classifier2 == y)

if count1 > count2:

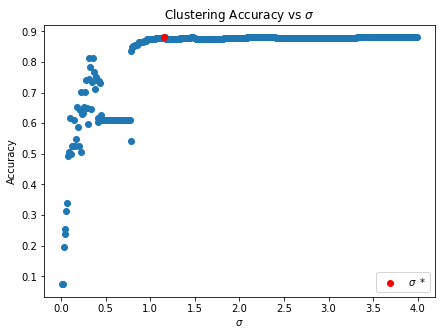
count = count1

else:

count = count2

accuracy = count / 435

## **Sec. 4. Computational Results**



(Figure 1) Scatter plot for clustering Accuracy vs . The value with the highest accuracy is colored red.

The optimal value I obtained is with an accuracy of .

## **Sec. 5. Summary and Conclusions**

We tested multiple values to find the optimal for spectral clustering.

## **Acknowledgments**

I would like to acknowledge the students in the AMATH 582/482 Discord who allowed me to have fruitful discussion about this assignment.

## **References**

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature

585, 357–362 (2020). <https://doi.org/10.1038/s41586-020-2649-2>

J. D. Hunter, "Matplotlib: A 2D Graphics Environment," in Computing in Science &

Engineering, vol. 9, no. 3, pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.