

## English

# SVM Machine Learning for EIS data analysis step-by-step user guide

Yue Rong, Agricultural and Biological Engineering, University of Florida

## 1. Software preparation

### 1.1. Download Anaconda installer

“Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment.” --[https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))

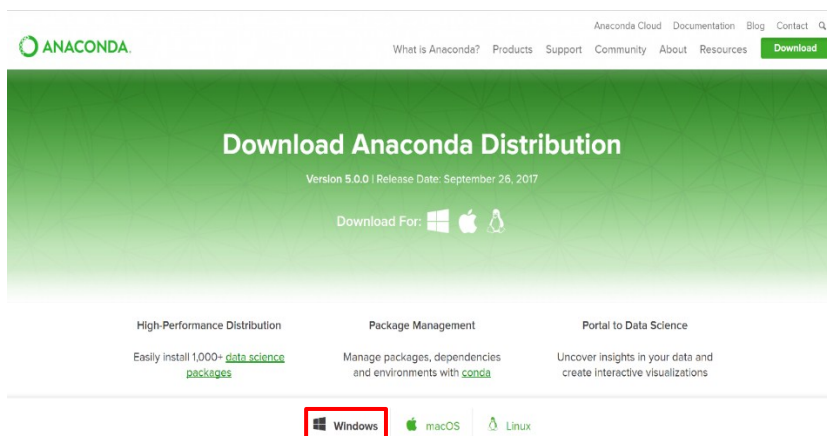
It includes all the packages and tools we need and is completely free.

It can be downloaded at:

<https://www.continuum.io/downloads>

There are different versions of Anaconda for Windows, macOS and Linux; Python 3.6 and 2.7. It will be about 2 Gb after installation and downloading all packages. We suggest Python 3.6.

a)



b)

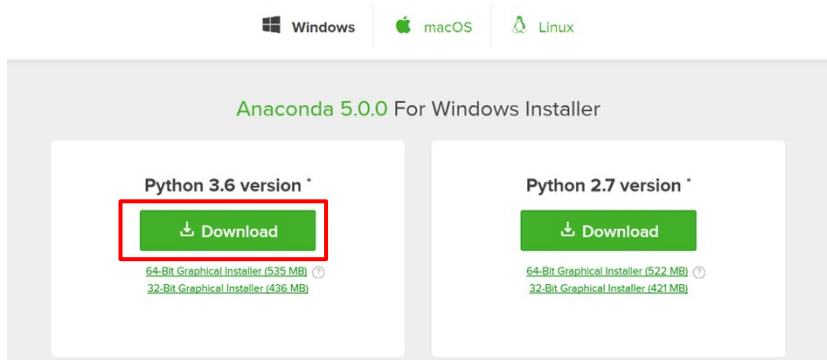


Figure U1. Download Anaconda

### 1.2. Install Anaconda

Double click the installer to launch

NOTE: If you encounter any issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you have installed for all users, uninstall Anaconda and re-install it for your user only and try again.

**Click Next.**

**Read the licensing terms and click “I Agree”**

Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.

**Select a destination folder to install Anaconda and click the Next button.**

NOTE: Install Anaconda to a directory path that does not contain spaces or unicode characters.

NOTE: Do not install as Administrator unless admin privileges are required.

**Choose whether to add Anaconda to your PATH environment variable.**

We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

**Choose whether to register Anaconda as your default Python 3.6.**

Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, you should accept the default and leave this box checked.

**Click the Install button.**

You can click Show Details if you want to see all the packages Anaconda is installing.

**Click the Next button**

After a successful installation, you will see the “Thanks for installing Anaconda” dialog box:

**Click the Finish button**

**After your install is complete, verify it by opening Anaconda Navigator**, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda.

**Anaconda Navigator**

If you had any issue installing the Anaconda, please visit <https://docs.anaconda.com/anaconda/install/> for help.

## 2. Getting started

### 2.1. Open Anaconda Navigator

Find Anaconda 3 in “Windows Start Menu” and click “Anaconda Navigator” to open

### 2.2. Click “Launch” to open “Jupyter Notebook”

*Figure 7 Lunch “Jupyter Notebook” in Anaconda Navigator*

It will open in your internet browser (for me it's a Chrome) and the default file location is under your system user folder. (for example C:\Users\Username)

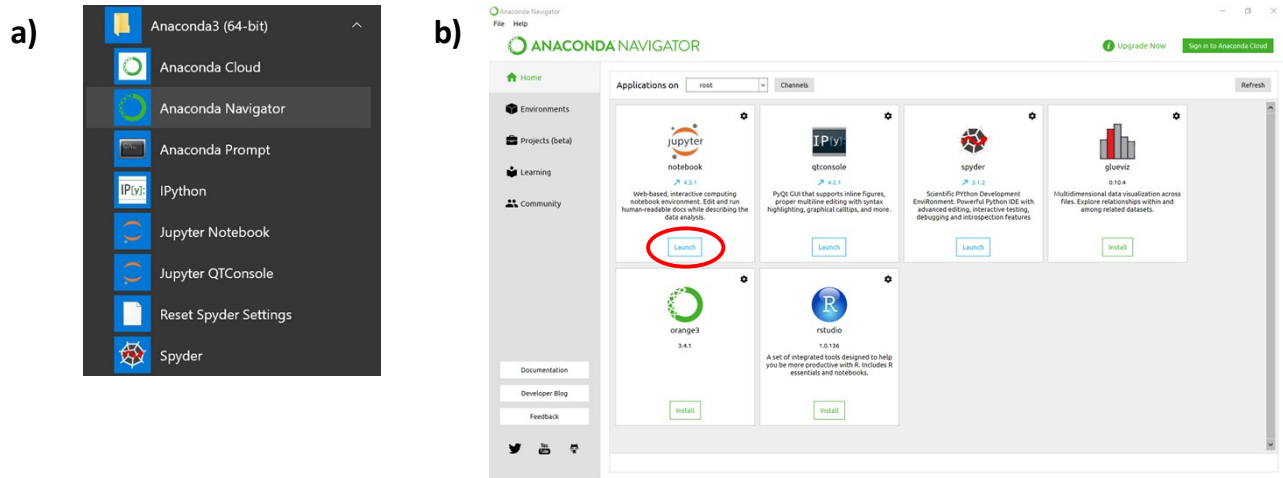


Figure U2. Open Jupyter Notebook

### 2.3. Run an example code

Download an example code from the link below and save it to “downloads” folder.

[http://scikit-learn.org/stable/downloads/plot\\_svm\\_nonlinear.ipynb](http://scikit-learn.org/stable/downloads/plot_svm_nonlinear.ipynb)

use the file browser to navigate and open the download source code above. It should look like Figure U3 below.

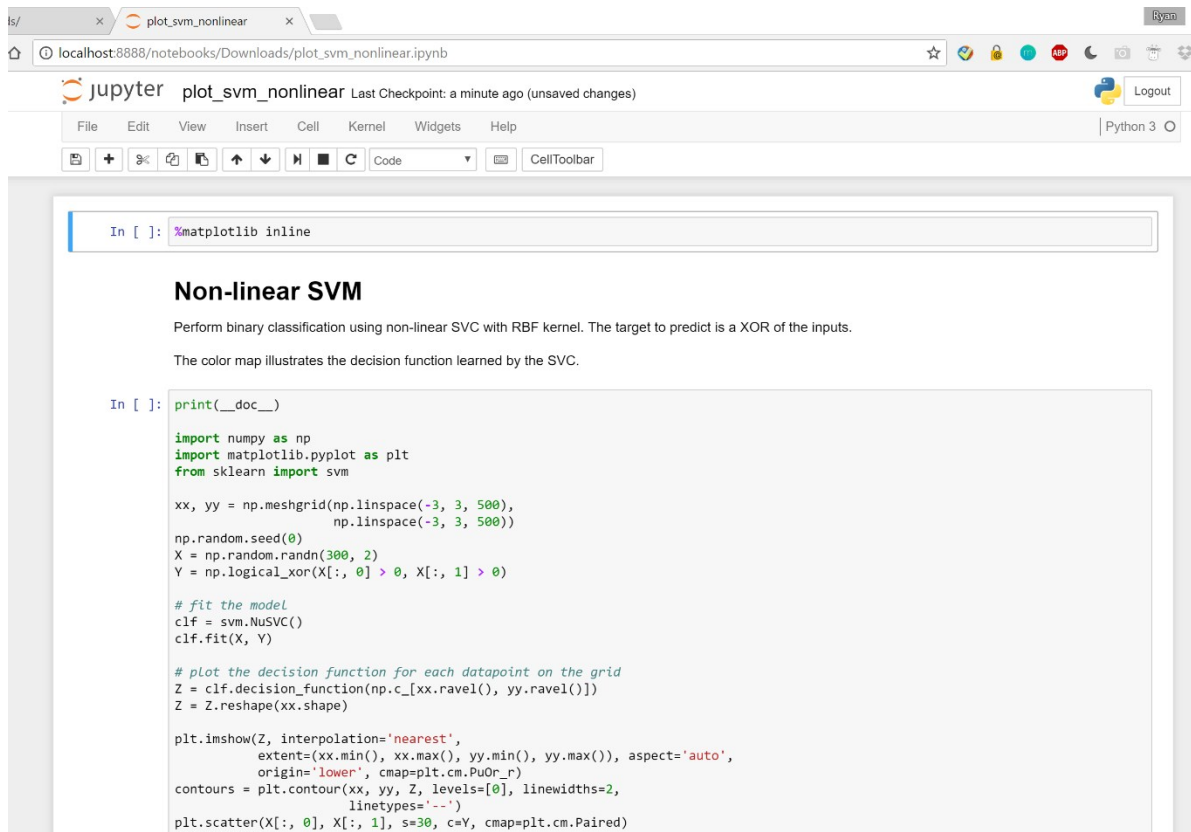



Figure U3. Example code for non-linear SVM

In “Cell” click “Run All” or run cells by clicking the  as shown in Figure U4.

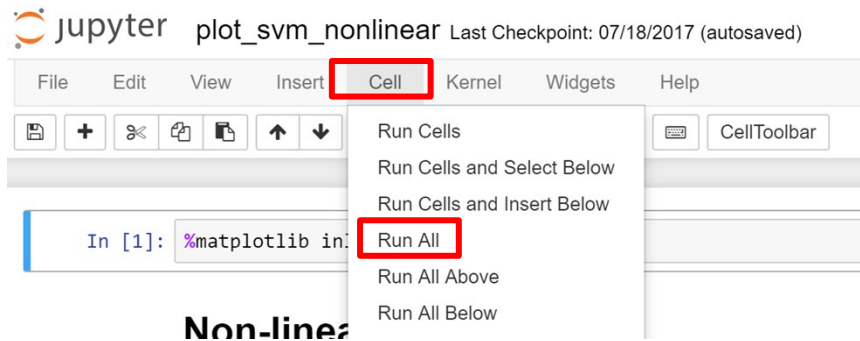


Figure U4. Run All cells to test the example code

The program should generate a plot similar to Figure U5.

```
plt.scatter(X[:, 0], X[:, 1], s=30, c=Y, cmap=plt.cm.Paired)
plt.xticks(())
plt.yticks(())
plt.axis([-3, 3, -3, 3])
plt.show()
```

Automatically created module for IPython interactive environment

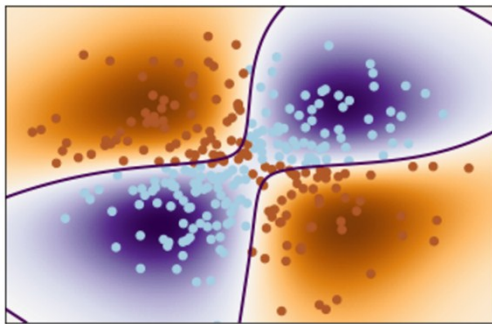


Figure U5. Expected output from example code

### 3. EIS data analysis with SVM machine learning

#### 3.1. Data preparation

Raw EIS data usually includes frequency with real impedance ( $Z'$ ) and imaginary impedance ( $Z''$ ) at that frequency as shown in Figure U6.

	A	B	C
	Frequency [Hz]	Z' _raw [Ohm]	Z'' _raw [Ohm]
0	99999.99984	258.054229	-7.637512
1	85769.58917	255.299691	-17.882566
2	73564.22411	278.492955	-18.88002
3	63095.73291	275.258838	-15.963164
4	54116.95083	289.347566	-28.062849
5	46415.88639	287.247927	-26.796648
6	39810.71513	286.058176	-32.585536
7	34145.48673	296.438676	-33.791812
8	29286.44376	301.039143	-34.728174
9	25118.8625	300.781714	-35.632715
10	21544.34515	307.305851	-40.781186
11	18478.4963	308.219676	-39.220288
12	15848.93039	312.550444	-44.296644
13	13593.56249	311.666707	-50.051774
14	11659.14268	315.616756	-54.547519
15	9999.998792	318.54106	-58.196081
16	8576.957885	322.457927	-63.004504
17	7356.421543	326.558636	-69.321829
18	6309.57253	330.949427	-77.043239
19	5411.69444	336.52842	-84.527607
20	4641.588089	343.795531	-92.286639
21	3981.071029	352.501566	-102.064143
22	3414.548269	359.595675	-112.278786
23	2928.644022	372.104547	-122.381571
24	2511.885947	385.386363	-134.632903

Figure U6. A part of representative raw EIS data.

Make sure all you EIS test are using the same frequency range then **create an excel sheet** as shown in Figure U7, where first column are names, second column are labels, each following column is one feature which is either real impedance or imaginary impedance at a frequency.

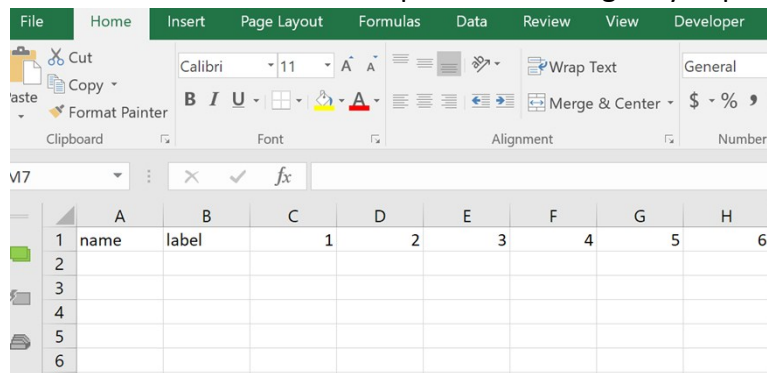


Figure U7. Excel template first step

Then, copy and transpose paste all real impedance (column B in Figure 12) to row 2 in Figure 13 starting at column C. (you may need to transpose paste the data in its original sheet then copy and paste to the template).

Next, using similar method as described above, copy all imaginary impedance (column C in Figure 12) to row 2 in Figure 13 starting at column CA (I have 76 frequencies in one EIS, I named row 1 from C as 1,2,3...76,77, to 152). Use your own naming system to give each EIS scan a unique name and put it in column A. Set a label (0 or 1 for binary classification) and put it in column B as shown in Figure U8.

	A	B	C	BZ	CA	CB	EW	EX
1	name	label	1	76	77	78	151	152
2	4B2G1	0	258.054229	1621.41	-7.63751	-17.8826	-873.914	-939.174

Figure U8. Pasted imaginary impedance into row 2 from column CA to EX (columns D-BY and CC-EV are hidden). Name and label added

Transfer all you EIS data that you are interested in classification (usually with similar experimental setting and a single variable) to the template and save it as \*.csv file

Figure U9. Example raw EIS dataset ready for SVM machine learning

This example file can be downloaded from:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

Each row is a data from one run of a sample (one EIS/CV scan)

The label column is used for supervised classification/regression. It could be binary or quantitate depends on your applications.

For example:

*In this dataset, “0” represent the blank/negative EIS results with no acetone in the samples; “1” represent the positive results with 5mM acetone concentration in the samples.*

*If you are testing glucose with 0, 1, 10 and 100 mM, you may label them as either “0,1,2,3” or “0,1,10,100” depending on doing classification or regression.*

Each column followed is called a “feature” that in EIS is the real or imaginary impedance at a certain frequency.

### 3.2. SVM code for EIS data

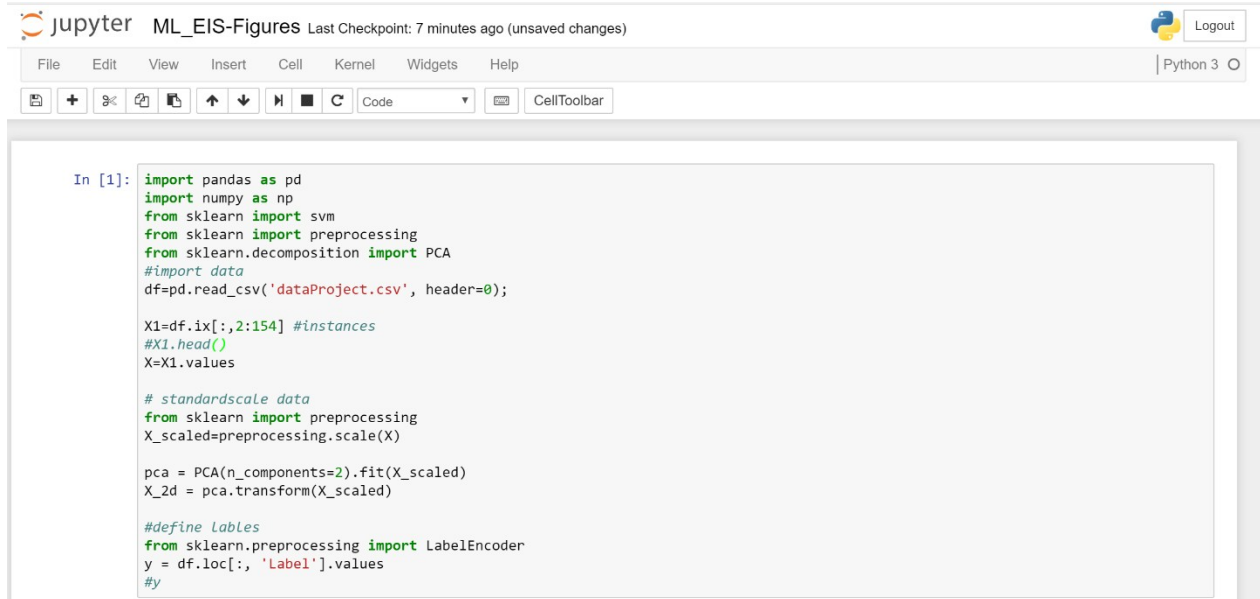
The code can be downloaded from:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

Opened it in “Jupyter notebook” and put the datafile prepared as above in the same folder.

Run the code by cells.

## 用 “Jupyter notebook”



```
In [1]: import pandas as pd
import numpy as np
from sklearn import svm
from sklearn import preprocessing
from sklearn.decomposition import PCA
#import data
df=pd.read_csv('dataProject.csv', header=0);

X1=df.ix[:,2:154] #instances
#X1.head()
X=X1.values

# standardscale data
from sklearn import preprocessing
X_scaled=preprocessing.scale(X)

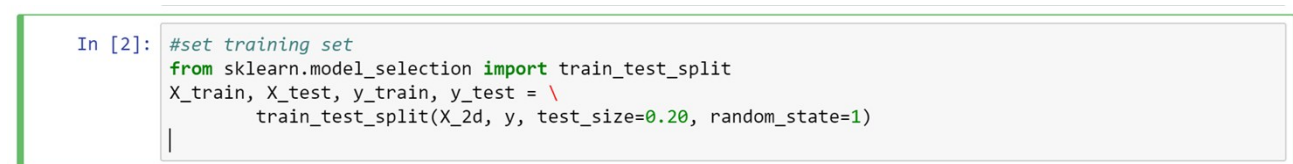
pca = PCA(n_components=2).fit(X_scaled)
X_2d = pca.transform(X_scaled)

#define Labels
from sklearn.preprocessing import LabelEncoder
y = df.loc[:, 'Label'].values
#y
```

Figure U10. reading EIS dataset and data preprocessing

This cell read the EIS dataset from 'dataProject.csv'

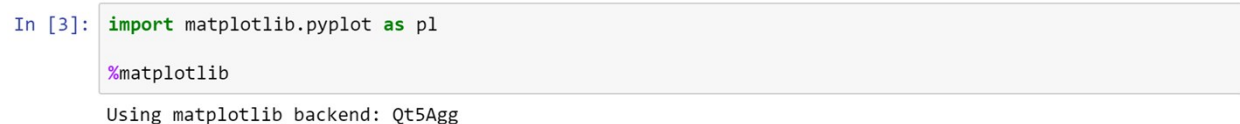
The data was scaled using `preprocessing.scale(X)`. And a principle component analysis (PCA) was performed to decompose the data into two dimensional.



```
In [2]: #set training set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = \
    train_test_split(X_2d, y, test_size=0.20, random_state=1)
```

Figure U11. Split data into training and testing sets.

EIS data were split into 80% training and 20% testing.



```
In [3]: import matplotlib.pyplot as plt

%matplotlib

Using matplotlib backend: Qt5Agg
```

Figure U12. Pyplot from matplotlib package was imported and figures are set to appear in a new window.

Figures plotting tools was loaded.

```

In [4]: #=====
#Fig 2
#=====
h = .02 # step size in the mesh
#C = 10 # SVM regularization parameter
svc = svm.SVC(kernel='linear').fit(X_train, y_train)
rbf_svc = svm.SVC(kernel='rbf').fit(X_train, y_train)
poly_svc = svm.SVC(kernel='poly').fit(X_train, y_train)
sigmoid_svc = svm.SVC(kernel='sigmoid').fit(X_train, y_train)

# create a mesh to plot in
x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                    np.arange(y_min, y_max, h))

# title for the plots
titles = ['SVC with linear kernel',
         'SVC with sigmoid kernel',
         'SVC with RBF kernel',
         'SVC with polynomial (degree 3) kernel']

for k, clf in enumerate((svc, sigmoid_svc, rbf_svc, poly_svc)):
    # Plot the decision boundary. For that, we will assign a color to each
    # point in the mesh [x_min, x_max]x[y_min, y_max].
    pl.subplot(2, 2, k + 1)
    pl.subplots_adjust(wspace=0.4, hspace=0.4)

    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])

    # Put the result into a color plot
    Z = Z.reshape(xx.shape)
    pl.contourf(xx, yy, Z, cmap=pl.cm.coolwarm, alpha=0.5)

    # Plot also the training points
    pl.scatter(X_2d[:, 0], X_2d[:, 1], c=y, cmap=pl.cm.coolwarm, s=20)

    #Plot score
    pl.text(0.6,0.9, ('Test accuracy: %.2F' % clf.score(X_2d, y)), transform=pl.gca().transAxes,
           bbox={'facecolor':'white'})

    pl.xlabel('Sepal length')
    pl.ylabel('Sepal width')
    pl.xlim(xx.min(), xx.max())
    pl.ylim(yy.min(), yy.max())
    pl.xticks(())
    pl.yticks(())
    pl.title(titles[k])

pl.show()

```

Figure U13. code cell for kernel selection

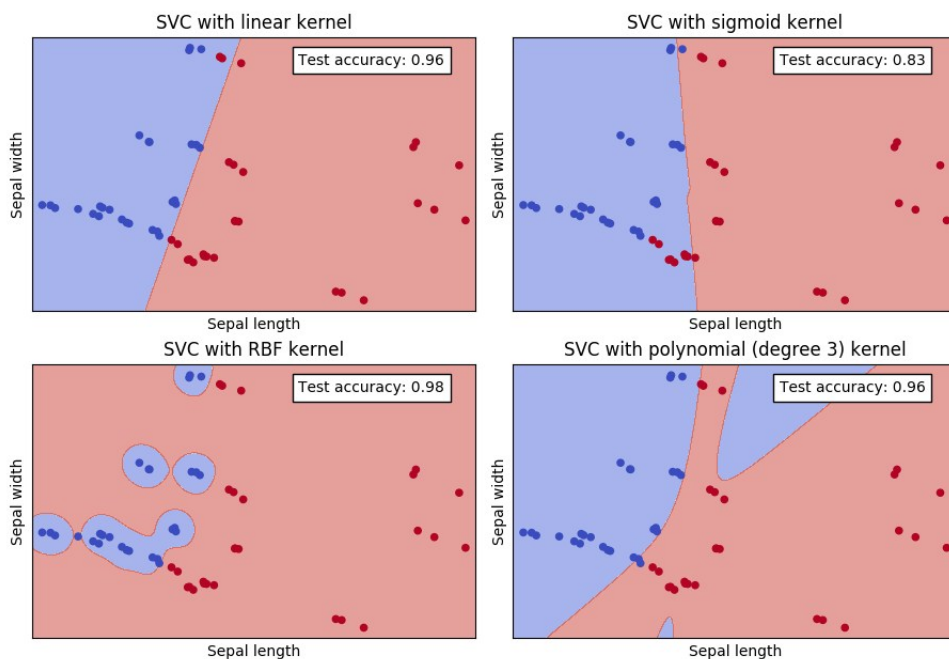


Figure U14. Result figure from the kernel selection code



```
In [6]: from sklearn.model_selection import cross_val_score

scores = cross_val_score(estimator=svm.SVC(kernel='rbf', gamma=0.01, C=10),
                        X=X_2d,
                        y=y,
                        cv=10,
                        n_jobs=10)
print('CV accuracy scores: %s' % scores)
print('CV accuracy: %.3f +/- %.3f' % (np.mean(scores), np.std(scores)))

CV accuracy scores: [ 0.66666667  1.          0.83333333  1.          1.          1.          1.          1.          1.          1.          ]
CV accuracy: 0.950 +/- 0.107
```

Figure U15. Cross validation and its result.

```
In [7]: from matplotlib.colors import Normalize

from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_iris
from sklearn.model_selection import StratifiedShuffleSplit
from sklearn.model_selection import GridSearchCV
%matplotlib

class MidpointNormalize(Normalize):
    def __init__(self, vmin=None, vmax=None, midpoint=None, clip=False):
        self.midpoint = midpoint
        Normalize.__init__(self, vmin, vmax, clip)

    def __call__(self, value, clip=None):
        x, y = [self.vmin, self.midpoint, self.vmax], [0, 0.5, 1]
        return np.ma.masked_array(np.interp(value, x, y))

C_range = np.logspace(-5, 8, 14)
gamma_range = np.logspace(-9, 4, 14)
param_grid = dict(gamma=gamma_range, C=C_range)
cv = StratifiedShuffleSplit(n_splits=10, test_size=0.2, random_state=12)
grid = GridSearchCV(SVC(), param_grid=param_grid, cv=cv)
grid.fit(X_train, y_train)

print("The best parameters are %s with a score of %0.2f"
      % (grid.best_params_, grid.best_score_))

# Now we need to fit a classifier for all parameters in the 2d version
# (we use a smaller set of parameters here because it takes a while to train)

C_2d_range = [1,10, 100, 1000,10000]
gamma_2d_range = [1e-4,1e-3, 1e-2,1e-1,1]
classifiers = []
for C in C_2d_range:
    for gamma in gamma_2d_range:
        clf = SVC(C=C, gamma=gamma)
        clf.fit(X_train, y_train)
        classifiers.append((C, gamma, clf))

Using matplotlib backend: Qt5Agg
The best parameters are {'C': 10.0, 'gamma': 0.01} with a score of 0.99
```

Figure U16. Finding the best parameters for RBF kernel and its results.

```

In [9]: import matplotlib.pyplot as plt
# plt.figure(figsize=(8, 6))
# xx, yy = np.meshgrid(np.linspace(-3, 3, 200), np.linspace(-3, 3, 200))
h = 1
x_min, x_max = X_train[:, 0].min() - 1, X_train[:, 0].max() + 1
y_min, y_max = X_train[:, 1].min() - 1, X_train[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                    np.arange(y_min, y_max, h))
for (k, (C, gamma, clf)) in enumerate(classifiers):
    # evaluate decision function in a grid
    Z = clf.decision_function(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    # visualize decision function for these parameters
    plt.subplot(len(C_2d_range), len(gamma_2d_range), k + 1)
    plt.subplots_adjust(wspace=0.3, hspace=0.3)
    plt.title("gamma=10-%d, C=10-%d" % (np.log10(gamma), np.log10(C)),
             size='medium')

    # visualize parameter's effect on decision function
    Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)
    plt.contourf(xx, yy, Z, cmap=plt.cm.coolwarm, alpha=0.5)
    # plt.pcolormesh(xx, yy, -Z, cmap=plt.cm.RdBu)
    plt.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap=plt.cm.coolwarm)
    # plt.scatter(X_test[:, 0], X_test[:, 1], y_test, cmap=plt.cm.RdBu_r)

    for i in range(0, X_train.shape[0]):
        if y_train[i] == 0:
            c1 = plt.scatter(X_train[i,0], X_train[i,1], c='r', marker='o', s = 20)
        elif y_train[i] == 1:
            c2 = plt.scatter(X_train[i,0], X_train[i,1], c='g', marker='+', s = 30)
    for i in range(0, X_test.shape[0]):
        if y_test[i] == 0:
            c3 = plt.scatter(X_test[i,0], X_test[i,1], c='b', marker='o', s = 20)
        elif y_test[i] == 1:
            c4 = plt.scatter(X_test[i,0], X_test[i,1], c='m', marker='+', s = 30)

    plt.xticks(())
    plt.yticks(())
    plt.axis('tight')

scores = grid.cv_results_['mean_test_score'].reshape(len(C_range),
                                                    len(gamma_range))

```

Figure U17. Code to visualize effect of parameters for RBF kernel.

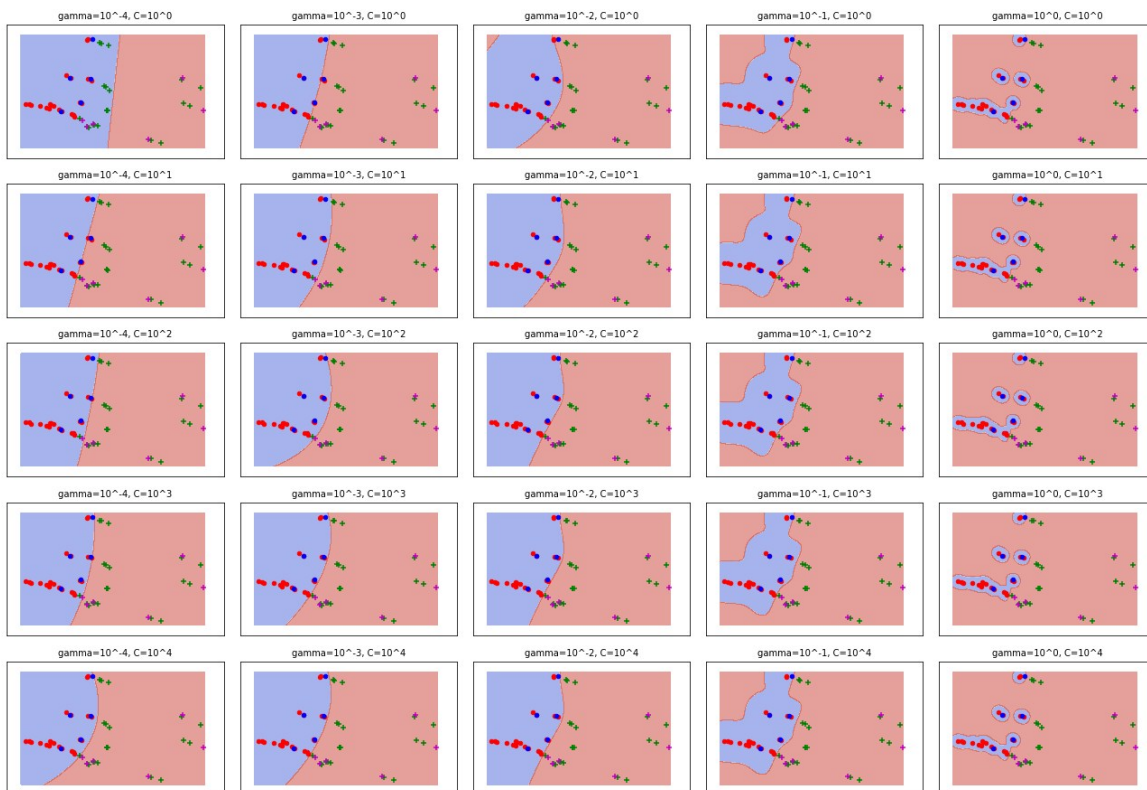


Figure U18. Effect of parameters for RBF kernel.

If you have your dataset prepared like this you may easily modify my code or the codes available online and test it out.

### **3.3. Scikit-Learn**

<http://scikit-learn.org/stable/index.html>

If you have time, checkout this site.

Most of my code and knowledge is based on the information and example from this open source project.

Good Luck!

10/23/2017



## Spanish

# Máquinas de soporte vectorial (SVM) en aprendizaje automático para el análisis de datos de espectroscopia de impedancia electroquímica (EIS): Guía de usuario paso a paso

Yue Rong, Departamento de Ingeniería Agrícola y Biológica, Universidad de la Florida, Gainesville -FL

## 4. Preparación del software

### 4.1. Descargar Anaconda

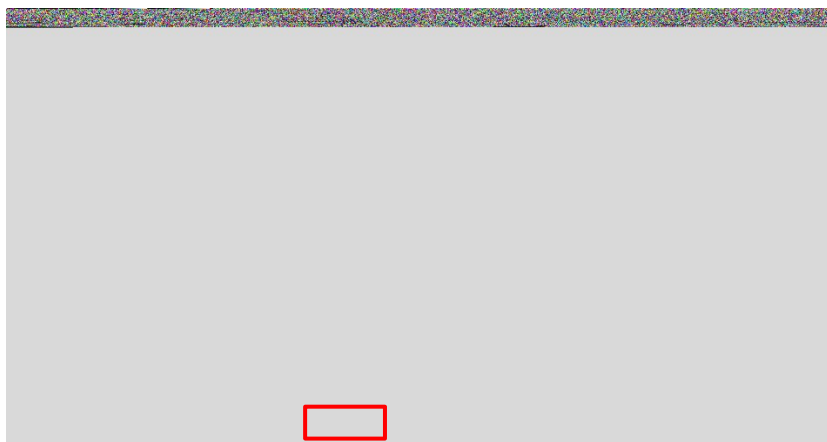
Anaconda es una distribución de código abierto gratuito de los lenguajes de programación Python y R para procesamiento de datos a gran escala, análisis predictivo y computación científica, que tiene como objetivo simplificar la administración e implementación de paquetes.

Este software incluye todos los paquetes y herramientas necesarios para nuestro propósito, y es completamente gratuito.

Se puede descargar en: <https://www.continuum.io/downloads> (ver figura U1)

Existen diferentes versiones de Anaconda para Windows, macOS y Linux; Python 3.6 y 2.7. Se requieren aproximadamente 2 GB para la descarga e instalación de todos los paquetes. Sugerimos la versión Python 3.6.

a)



b)

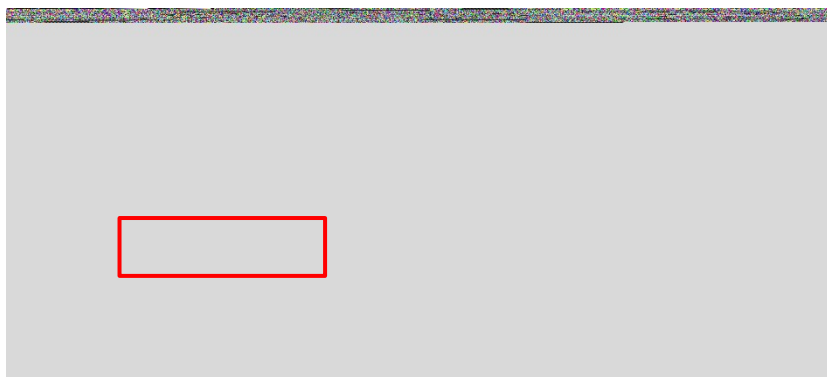


Figure U1. Descarga de Anaconda Distribution

## PARA USUARIOS DE WINDOWS:

### 4.2. Instalar Anaconda

#### 1.2.1 Haga doble clic en el instalador para iniciar

NOTA: Si encuentra algún problema durante la instalación, deshabilite temporalmente su software antivirus, y vuelva a habilitarlo después de que finalice la instalación de Anaconda.

#### 1.2.2 Haga clic en "Next"

#### 1.2.3 Lea los términos de la licencia y haga clic en "I Agree"

1.2.4 Seleccione la opción "Just Me" a menos que esté instalando el programa para todos los usuarios del Sistema (lo cual requiere privilegios de administrador de Windows) y haga clic en Siguiente. En caso de que se haya instalado para todos los usuarios, desinstale Anaconda y vuelva a instalarlo para un solo usuario.

#### 1.2.5 Seleccione una carpeta de destino para instalar Anaconda y haga clic en el botón "Next".

NOTA: Instale Anaconda en una ruta de directorio que no contenga espacios ni caracteres Unicode. No instale como administrador a menos que se requieran privilegios de administrador.

#### 1.2.6 Seleccione la casilla "Register Anaconda as my default Python 3.6".

NOTA: A menos que planea instalar y ejecutar varias versiones de Anaconda, o varias versiones de Python, debe aceptar el valor predeterminado y dejar esta casilla marcada. Recomendamos no agregar Anaconda a la variable de entorno PATH, ya que esto puede interferir con otros programas. En lugar de esto, use el software Anaconda abriendo Anaconda Navigator o Anaconda Prompt desde el Menú de Inicio.

#### 1.2.7 Haga clic en el botón "Install"

Nota: Puede hacer clic en Mostrar detalles si desea ver todos los paquetes que Anaconda está instalando.

1.2.8 Una vez completada la instalación, haga clic en el botón "Next". Después de una instalación exitosa, verá el cuadro de diálogo "thanks for installing Anaconda3".

NOTA: En caso de que el instalador le sugiera la instalación de "Visual Studio Code", puede ignorar este procedimiento haciendo clic en el botón **Skip**. Visual Studio Code es un editor de código de plataforma cruzada gratuito y de fuente abierta que provee un excelente soporte para la edición de código Python. Sin embargo la instalación requiere privilegios de administrador y conexión a internet.

#### 1.2.9 Haga clic en el botón "Finish"

Una vez completada la instalación, verifíquela abriendo Anaconda Navigator, que es un programa que se incluye con Anaconda. Para ello, desde el menú Inicio de Windows, seleccione el acceso directo Anaconda Navigator. Si se abre la aplicación ANACONDA NAVIGATOR, ha instalado el software correctamente.

## Anaconda Navigator

Si tuvo algún problema al instalar Anaconda, visite <https://docs.anaconda.com/anaconda/install/> para obtener ayuda.

## 5. Comenzando

### 5.1. Abrir el Navegador de Anaconda

Para abrir el navegador, seleccione el acceso directo Anaconda Navigator del menu de Windows, o alternativamente busque la carpeta Anaconda 3 en el menu de Windows y haga clic en "Anaconda Navigator" (ver figura U2)

### 5.2. Abra la aplicación " Jupyter Notebook" haciendo clic en el boton "launch"

Esta aplicacion se abrirá en su navegador de Internet. La ubicación predeterminada del archivo se encuentra en la carpeta de usuario de su sistema. (por ejemplo C: \ Users \ Username)

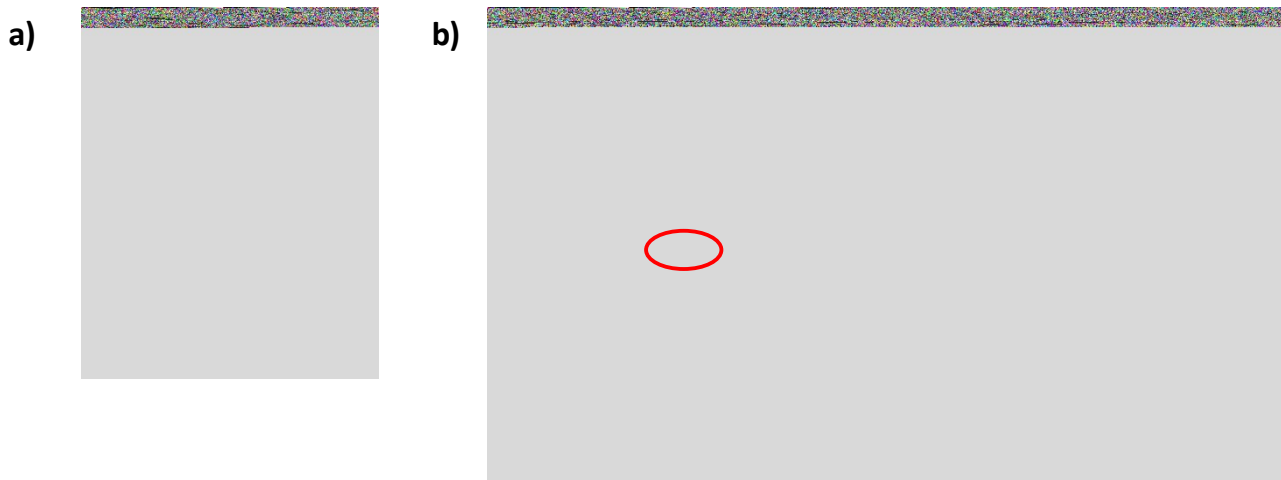


Figure U2. Interfaz gráfica de Anaconda Navigator. Se señala el boton de acceso de la aplicación Jupyter

### 5.3. Ejecute un código de ejemplo

**2.3.1** Abra una pestaña paralela en el navegador y dirijase al sitio web de scikit learn:


[http://scikit-learn.org/stable/auto\\_examples/svm/plot\\_svm\\_nonlinear.html](http://scikit-learn.org/stable/auto_examples/svm/plot_svm_nonlinear.html)

**2.3.2** Dirijase a la parte inferior derecha de la pagina, y haga clic en el boton "Download Jupyter Notebook: plot\_svm\_nonlinear.ipynb".

**2.3.3** Regrese a la aplicacion de jupyter y abra la carpera "downloads". Haga clic sobre el archivo "plot\_svm\_nonlinear.ipynb". Debería verse como la Figura U3 a continuación.



Figure U3. Código de ejemplo para SVM no lineal

**2.3.4** En "Cell", haga clic en "Run All" o ejecute celdas haciendo clic en el icono  que se muestra en la Figura U4.

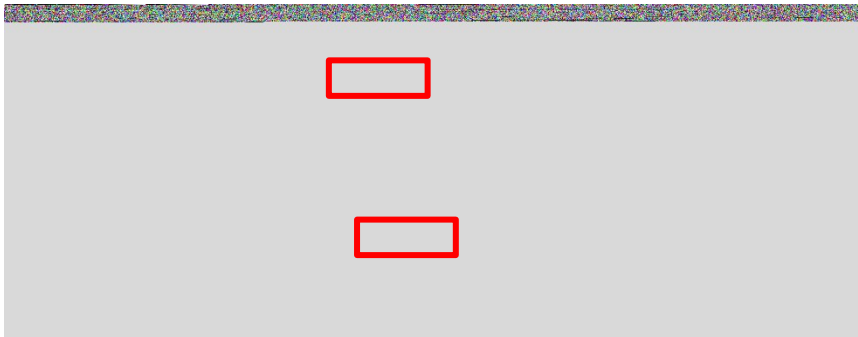


Figure U4. Ejecute todas las celdas para probar el código de ejemplo

El programa debe generar un diagrama similar a la Figura U5.



```
plt.scatter(X[:, 0], X[:, 1], s=30, c=Y, cmap=plt.cm.Paired)
plt.xticks(())
plt.yticks(())
plt.axis([-3, 3, -3, 3])
plt.show()
```

Automatically created module for IPython interactive environment

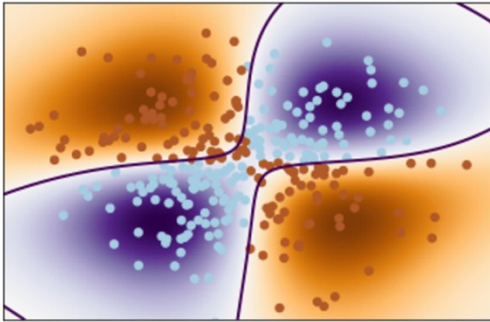


Figure U5. Salida esperada del código de ejemplo

## 6. Análisis de datos EIS con aprendizaje automático SVM

### 6.1. Preparación de datos

Los datos crudos de EIS usualmente incluyen tres variables: frecuencia, impedancia real ( $Z'$ ), e impedancia imaginaria ( $Z''$ ), como se muestra en la Figura U6.

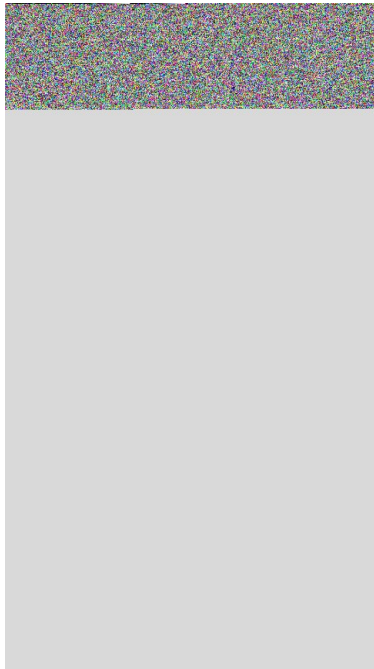


Figure U6. Ejemplo de datos crudos en un experimento de EIS.

**3.1.1** Asegúrese de que todas las pruebas de EIS estén usando el mismo rango de frecuencia.

**3.1.2** Prepare una plantilla de Excel como se muestra en la Figura U7, donde la primera columna son nombres, la segunda columna son etiquetas, y cada columna siguiente (i.e. E, F, G, H,...) son las variable de respuesta obtenidas directamente de los datos crudos, es decir impedancia real, e impedancia imaginaria a una frecuencia dada.



Figure U7. Plantilla de Excel primer paso.

**3.1.3** Copie y pegue todos los datos de impedancia real (columna B en la Figura 12) en la fila 2, comenzando en la columna C. (es posible que tenga que transponer los datos en su hoja original y luego copiar y pegar en la plantilla).

**3.1.4** Usando un método similar al descrito arriba, copie todos los resultados de impedancia imaginaria (columna C en la Figura 12) y peguelos en la fila 2 comenzando en la siguiente columna disponible (en ejemplo mostrado en la figura U8 es la columna CA porque cada variable de respuesta contenía 76 datos).

**3.1.5** Use su propio sistema de asignación de nombres para darle a cada corrida de EIS un nombre único y colóquelo en la columna A.

**3.1.6** Establezca una etiqueta (0 o 1 para la clasificación binaria) y colóquela en la columna B como se muestra en la Figura U9.



Figure U8. Impedancia imaginaria en la fila 2 de la columna CA a la columna EX (las columnas D-BY y CC-EV están ocultas). Nombre y etiqueta añadidos

**3.1.7** Transfiera todos los datos de EIS que le interese clasificar (generalmente con una configuración experimental similar y variable única) a la plantilla de excel y guárdelo con la extensión “.csv ”

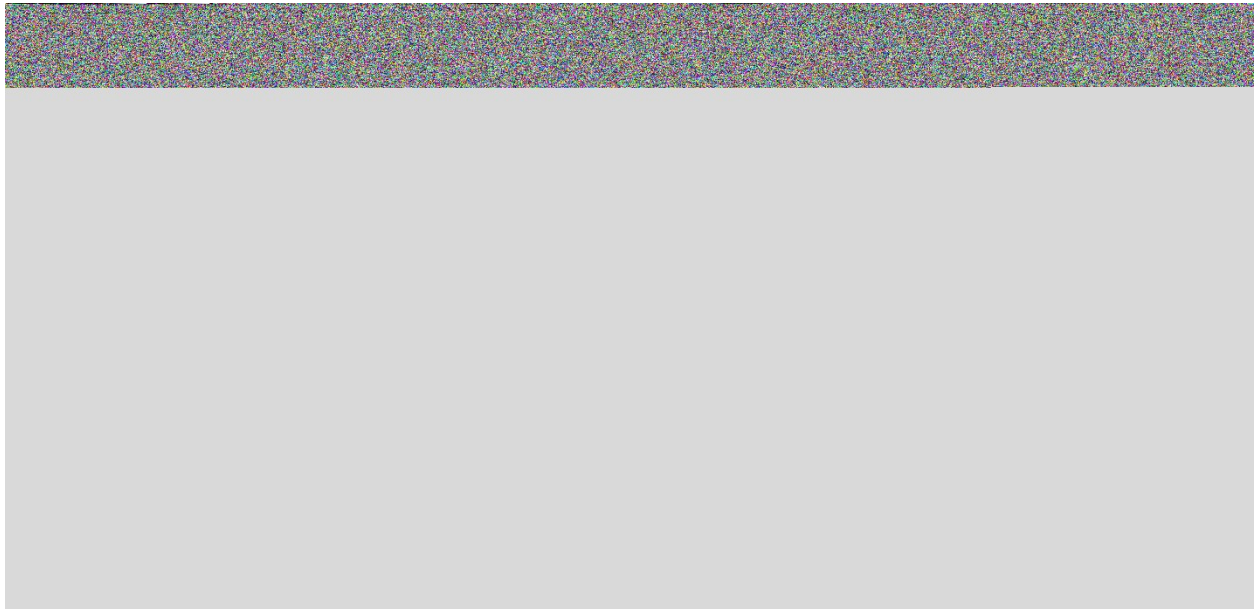


Figure U9. Ejemplo de conjunto de datos EIS listo para SVM

El archivo del ejemplo se puede descargar desde:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

NOTA: Cada fila son datos correspondientes a una corrida experimental (un barrido de EIS/CV). La columna de etiqueta se usa para la clasificación/regresión supervisada. Puede ser binario (como en este ejemplo) o cuantitativo, dependiendo de la aplicación de interés. Por ejemplo, en este conjunto de datos, "0" representa los resultados de EIS en la muestra blanco (i.e., control negativo, sin acetona); "1" representa los resultados positivos de las muestras con concentración de acetona mayor a 5 mM. Cada columna seguida se denomina "característica" que en el caso de datos de EIS es la impedancia real o imaginaria a una determinada frecuencia.

## 6.2. Código SVM para datos EIS

**3.2.1** Descargue el código. El código se puede descargar desde:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

**3.2.2** Guarde el archivo de código y el archivo de datos preparado con el procedimiento 3.1 en la misma carpeta del ordenador.

**3.2.3** Abra la aplicación "Jupyter notebook" (procedimiento 2.2)

**3.2.4** Desde la aplicación abra el archivo del código (ver figura U10)

**3.2.5** En la tercera línea del código incluya el nombre de su archivo de datos en lugar de "dataProject.csv"

**3.2.6** Ejecute el código por celdas (ver paso 2.3.4 y ver figuras U11 a U16).



Figure U10. Leer el conjunto de datos EIS y el preprocesamiento de datos

NOTA: Este programa esta diseñado para leer el conjunto de datos de EIS del archivo de datos preparados segun el procedimiento 3.1 (para el ejemplo mostrado es el archivo 'dataProject.csv'). Los datos se escalaron utilizando `preprocessing.scale (X)`. Y se realizó un análisis de componentes principales (PCA) para descomponer los datos en dos dimensiones.



Figura U11. División de los datos en los conjuntos de entrenamiento y prueba

NOTA: En este codigo los datos de EIS se dividieron en 80% de entrenamiento y 20% de prueba.



Figure U12. Pyplot del paquete matplotlib se importó y las figuras se configuraron para aparecer en una nueva ventana.

Las figuras trazando herramientas se cargaron.



Figure U13. Celda de código para la selección del kernel

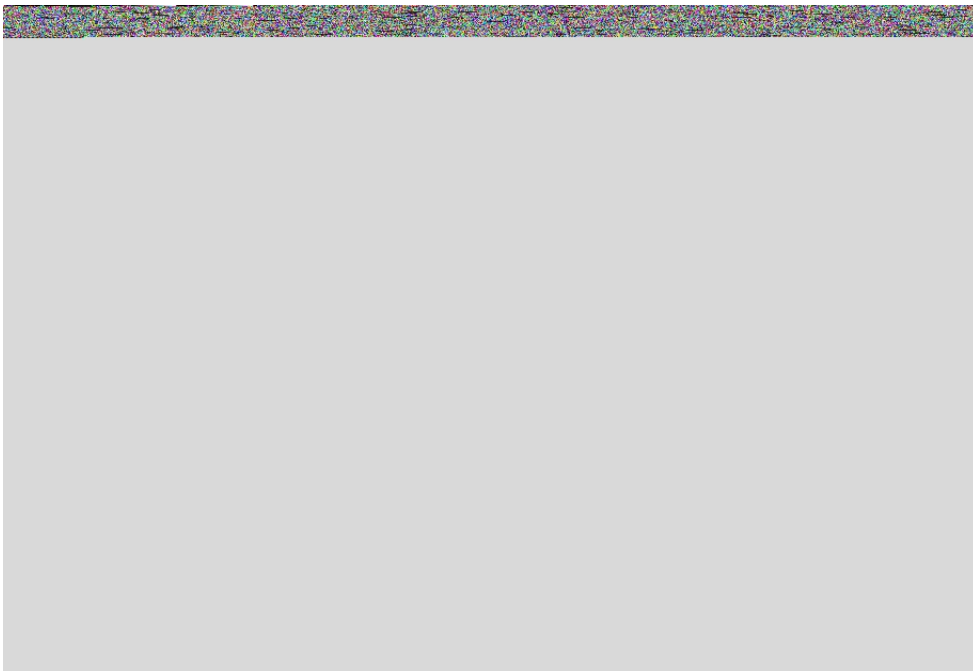


Figure U14. Resultado del código de selección del kernel



Figure U15. Validación cruzada

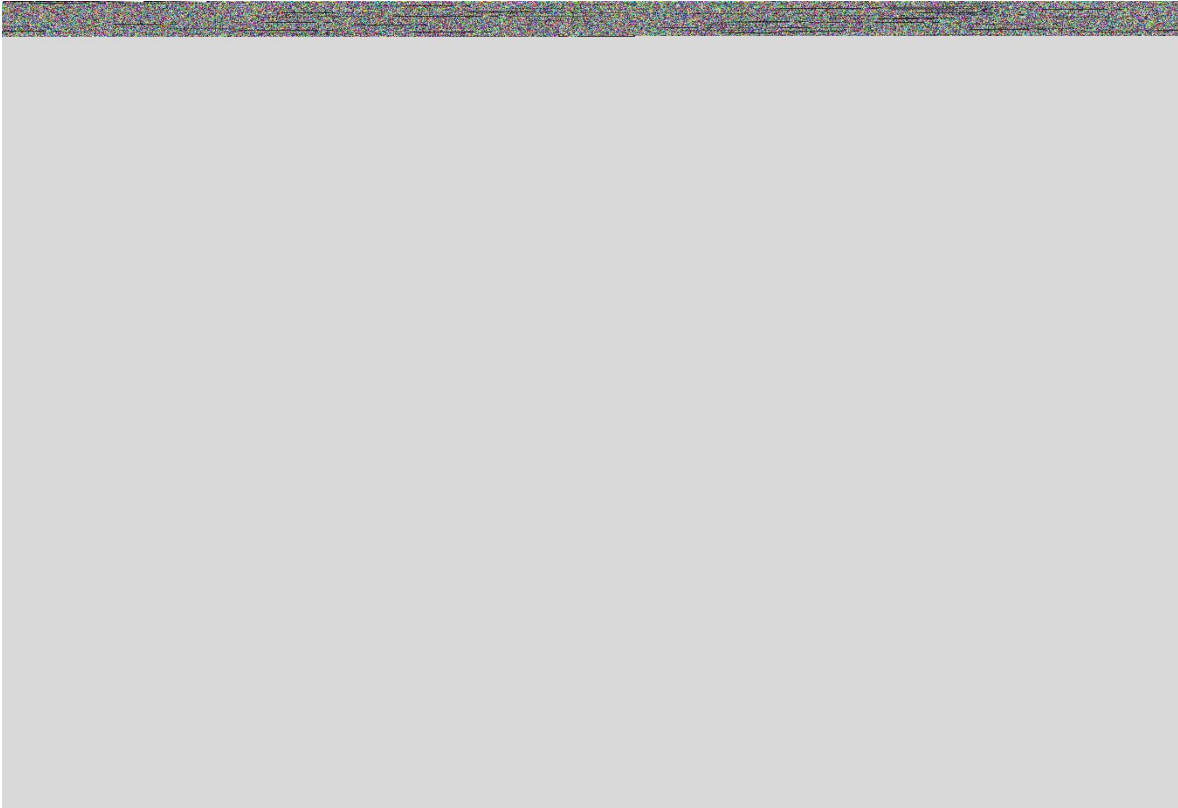


Figure U16. Busqueda de los mejores parámetros para kernel RBF.

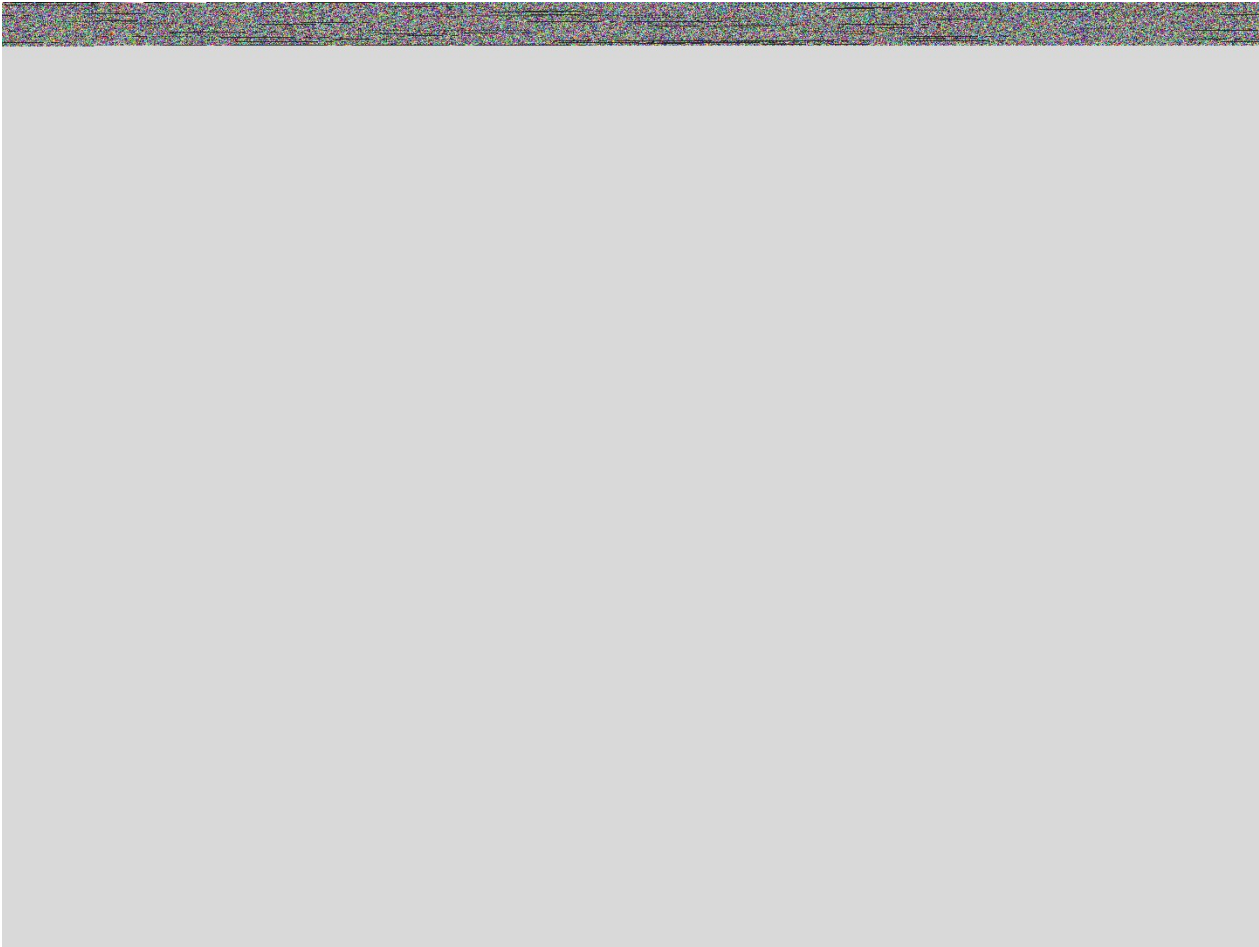


Figure U17. Código para visualizar el efecto de los parámetros para el kernel RBF.

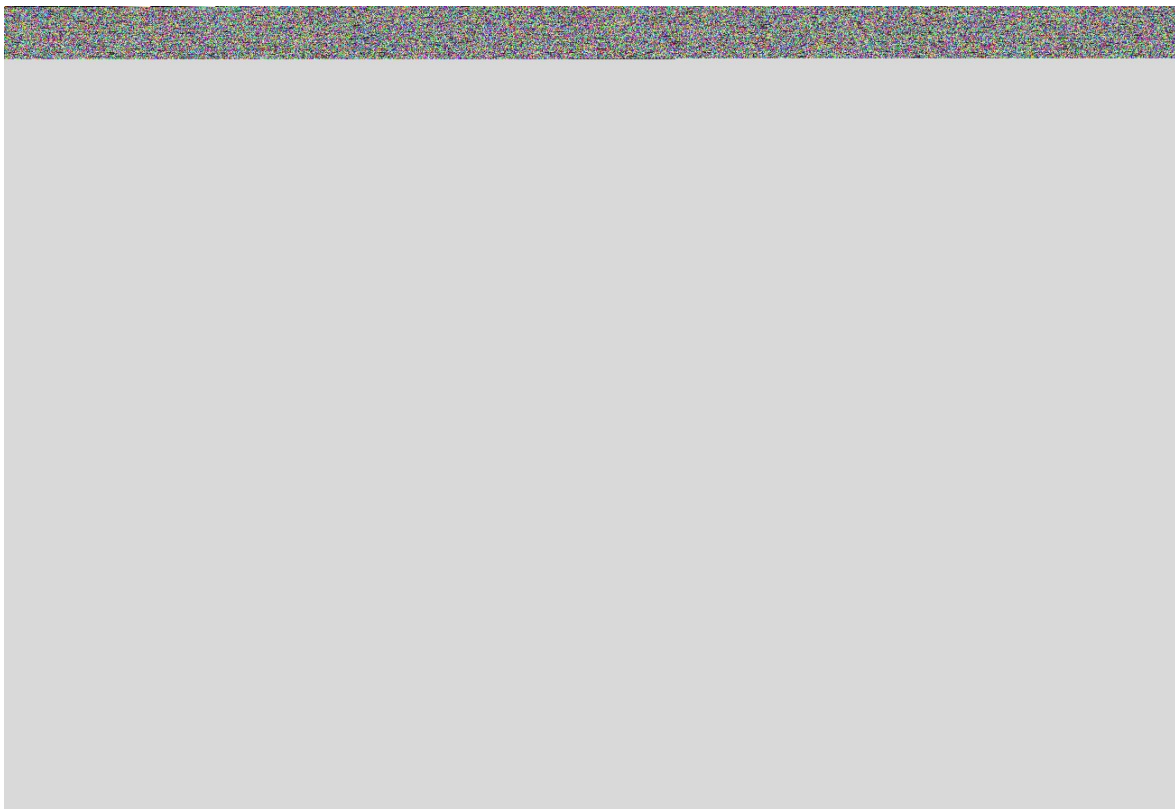


Figure U18. Efecto de los parámetros para el kernel RBF.

Si tiene su conjunto de datos preparado de esta manera, puede modificar fácilmente mi código o los códigos disponibles en línea y probarlo.

### **6.3. Scikit-Learn**

Si tiene tiempo, visite el sitio web de la organización SciKit:

<http://scikit-learn.org/stable/index.html>

La mayor parte de mi código y conocimiento se basa en la información y el ejemplo de este proyecto de código abierto.

Buena suerte,

Yue Rong

10/23/2017



## Portuguese

# Aprendizado de Máquina SVM para análise de resultados de EIS: guia do usuário passo-a-passo

Yue Rong, Engenharia Biológica e Agrícola, Universidade da Florida

## 7. Preparação do Software

### 7.1. Baixar o arquivo instalador Anaconda

“Anaconda é uma distribuição de código aberto freemium para as linguagens de programação Python e R que é utilizada para o processamento de dados em larga escala, análise preditiva, e computação científica, com o objetivo de simplificar lançamento e manutenção de sistemas.” --

[https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))

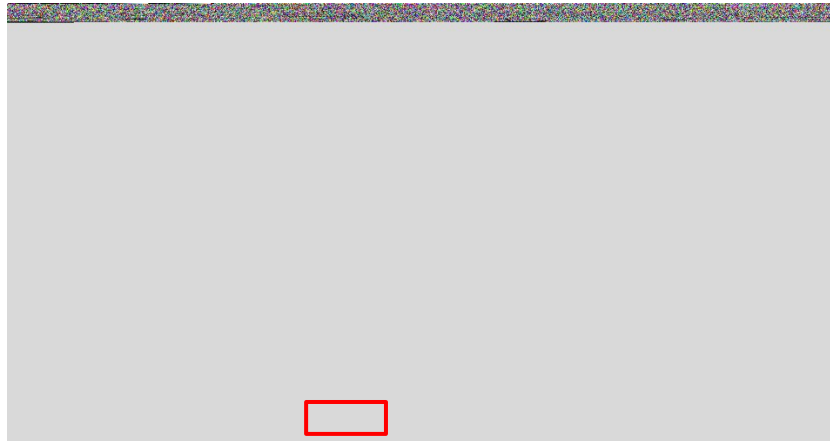
O instalador inclui todas as ferramentas e sistemas que nos precisamos e é totalmente grátis.

O instalador pode ser baixado no endereço abaixo:

<https://www.continuum.io/downloads>

Existem versões diferentes do Anaconda para Windows, macOS e Linux; Python 3.6 e 2.7. O que corresponde à aproximadamente 2 Gb após instalação de todos os sistemas. Nós sugerimos Python 3.6.

a)



b)

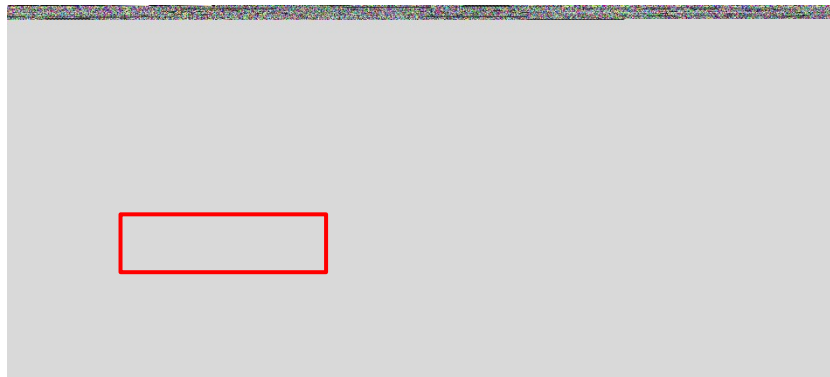


Figura U1. Baixar Anaconda.

### 7.2. Instalação do Anaconda

## **Clique duas vezes o instalador para iniciar a instalação**

NOTA: se você encontrar problemas durante a instalação, temporariamente desarme o seu software de anti-virus durante a instalação, e depois retorne a utilizá-lo uma vez que a instalação finalize. Se você selecionou para instalar para todos os usuários, desinstale o Anaconda, e reinstale para um usuário somente, e tente novamente.

## **Clique Avançar (“Next”)**

### **Leia os termos da licença (License Terms) e clique “Eu concordo” (“I agree”)**

Selecione a instalação para “somente você” (“Just me”) ao menos que você esteja instalando para todos os usuários (o que requer privilégios de administrador no Windows) e clique Avançar (“Next”).

### **Selecione o destino da pasta (“folder destination”) para instalação do Anaconda e clique o botão Avançar (“Next”).**

NOTA: Instale o Anaconda em um caminho de diretório que não contenha espaços ou caracteres unicódigos.

NOTA: Não instale como Administrador a não ser que sejam requeridos privilégios administradores.

### **Escolha se você prefere adicionar Anaconda a sua variável ambiental PATH.**

Nós recomendamos não adicionar Anaconda a variável ambiental PATH, uma vez que isto pode interferir com outros software. Ao invés disso, use Anaconda software abrindo o navegador Anaconda (Anaconda Navigator) ou o Anaconda Prompt no menu iniciar.

### **Escolha se você prefere registrar Anaconda como o seu destino predefinido (“default”) do Python 3.6.**

Ao menos que você planeje em instalar ou executar várias versões do Anaconda, ou múltiplas versões de Python, você deveria aceitar o destino predefinido (“default”) e deixar esta opção selecionada.

## **Clique o botão de instalação (“Install”).**

Você pode clicar “Mostre detalhes” (“Show Details”) se você quiser ver todos os pacotes de programas o Anaconda está instalando.

## **Clique o botão Avançar (“Next”)**

Após a instalação bem sucedida, você verá a caixa de diálogo “Obrigado por instalar Anaconda” (Thanks for installing Anaconda”):

## **Clique no botão Finalizar (“Finish”)**

**Após a instalação estiver finalizada, verifique se funcionou ao tentar abrir o Navegador Anaconda (“Anaconda Navigator”),** um programa que está incluído com o Anaconda: no seu menu iniciar do Windows, selecione o menu de atalho Navegador Anaconda (“Anaconda Navigator”). Se o Navegador abrir, você instalou corretamente o Anaconda.

## **Navegador Anaconda (“Anaconda Navigator”)**

Se você encontrar qualquer problema na instalação do Anaconda, por favor visite o site <https://docs.anaconda.com/anaconda/install/> para ajuda.

## 8. Introdução

### 8.1. Abrir o Navegador Anaconda (“Anaconda Navigator”)

Localize Anaconda 3 no “menu iniciar do Windows” e clique “Navegador Anaconda” (Anaconda Navigator”) para abrir.

### 8.2. Clique “Iniciar Programa” (“Launch”) para abrir “Livro de Júpiter” (“Jupyter Notebook”)

Isto irá abrir no seu navegador de internet (no meu caso é o Chrome) e a localização predefinida do artigo está no sistema de sua pasta de arquivos. (por exemplo: C:\Usuários\Nome do Usuário).

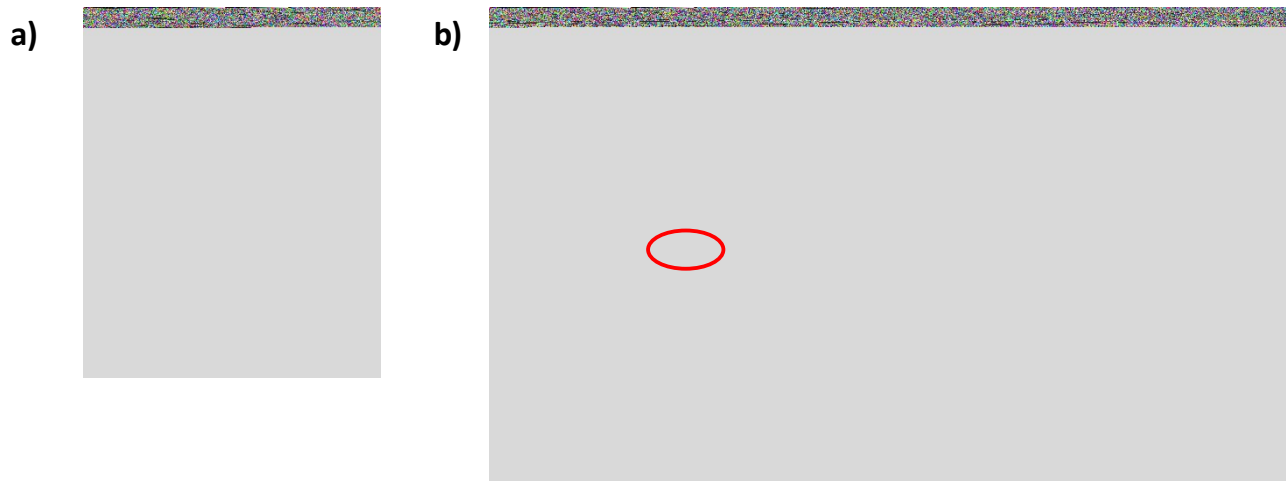


Figura U2. Abrir Livro de Júpiter (“Jupyter Notebook”).

### 8.3. Execute um código de exemplo

Abaxe um código de exemplo do link abaixo e salve na pasta de “downloads” (“downloads folder”).


[http://scikit-learn.org/stable/downloads/plot\\_svm\\_nonlinear.ipynb](http://scikit-learn.org/stable/downloads/plot_svm_nonlinear.ipynb)

Use o navegador de artigos para navegar e abrir o código de exemplo acima. Isto deve se assemelhar a Figure U3 abaixo.

*Figure 7 Execute “Livro de Júpiter”( “Jupyter Notebook”) no Navegador Anaconda (“Anaconda Navigator”)*



Figura U3. Código de Exemplo para o SVM não-linear.

No ícone “Celúla” (“Cell”) clique “Execute tudo” (“Run All”) ou execute células clicando no ícone , como ilustrado na Figura U4.

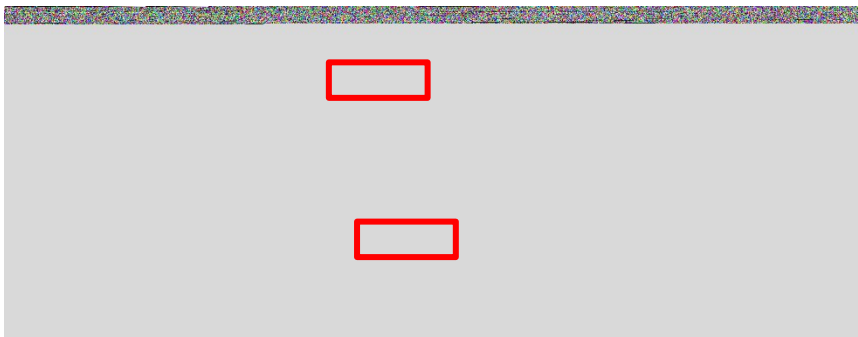


Figura U4. Execute todas (“Run All”) as células para testar o código de exemplo.

O programa deverá gerar um gráfico que se assemelha a Figura U5.



Figura U5. Resultado esperado para o código de exemplo.

## 9. Análise de dados de EIS utilizando o Aprendizado de Máquina SVM

### 9.1. Preparação dos Dados

Dados crus de EIS normalmente incluem frequência com impedância real ( $Z'$ ) e impedância imaginária ( $Z''$ ) daquela frequência em particular, como demonstrado na Figura U6.

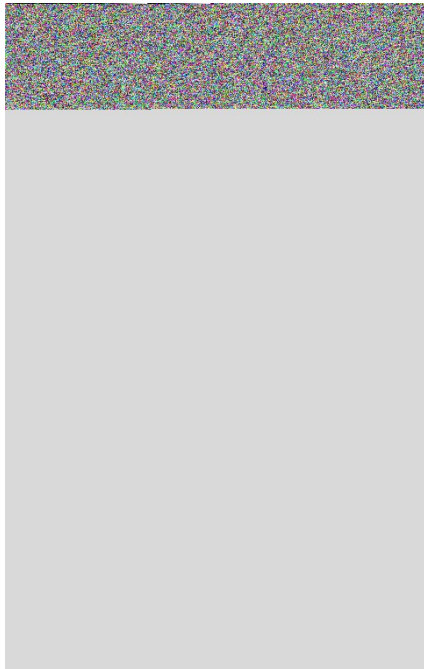


Figura U6. Uma porção representativa dos dados de EIS.

Confirme que todos os testes de EIS tem a mesma faixa de frequência, em seguida **crie uma planilha de excel**, como demonstrado na Figure U7, onde a primeira coluna são os nomes, a segunda coluna são os rótulos, e cada seguinte coluna é uma propriedade que corresponde a impedância real ou impedância imaginária de uma frequência em particular.



Figura U7. Planilha modelo no Excel - primeiro passo.

Em seguida, copie e transpõe todas as impedâncias reais (coluna B na Figure U6) para a linha 2 na Figura U7, iniciando na coluna C. (Você talvez precisará de tranpostar os dados na planilha original para depois copiar e colar na planilha modelo).

Em seguida, usando um método similar ao descrito acima, copie todas as impedâncias imaginárias (Coluna C na Figura U6) para linha 2 na Figure U7 iniciando na coluna CA (Eu tenho 76 frequências em um EIS, então eu nomeei a linha 1 de coluna C como 1, 2, 3...76,77, to 152). Utilize o seu próprio sistema de nomenclatura que proporcione cada escaneamento de EIS um nome único e coloque na coluna A. Defina um rótulo (0 ou 1 para classificação binária) e coloque na coluna B, como ilustrado na Figure U8.



Figura U8. Impedância imaginária colada na linha 2 originalmente da coluna CA até EX (coluna D-BY e CC-EV estão escondidas). Nome e rótulos foram adicionados.

Copie e cole todos os seus dados de EIS que você esteja interessado na classificação (normalmente com condições experimentais similares e uma única variável) para a planilha modelo e salve como artigo \*.csv.



Figura U9. Exemplo de dados crús prontos para o apredizando de máquina SVM.

Este exemplo de arquivo pode ser baixado por este link abaixo:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

Cada linha contém os dados de um teste de uma amostra (um escaneamento EIS ou Voltamograma Cíclico)

A coluna de rótulo é usada para classificação/regressão supervisionada. Esta coluna pode ser binária ou quantitativa dependendo da suas aplicações.

*Por exemplo:*

*Para estes dados, "0" representa os resultados de EIS das amostras brancas/negativas, sem acetona nas amostras; "1" representa os resultados positivos com concentração de 5 mM de acetona nas amostras.*

*Se você está testando glicose com 0, 1, 10 e 100 nM, você talvez queira rotular as amostras como "0, 1, 2, 3" ou "0, 1, 10, 100" dependendo se você está fazendo classificação ou regressão.*

Cada coluna em seguida é chamada de "propriedades" que no EIS são a impedância real ou impedância imaginária para uma frequência em particular.

## 9.2. Código do SVM para dados de EIS

O código pode ser baixado usando o link abaixo:

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

Abra o código no "Livro de Júpiter" ("Jupyter notebook") e coloque o arquivo de dados preparado como indicado acima na mesma pasta.

Execute o código usando "celúla" ("cells").

“Livro de Júpiter” (“Jupyter notebook” )

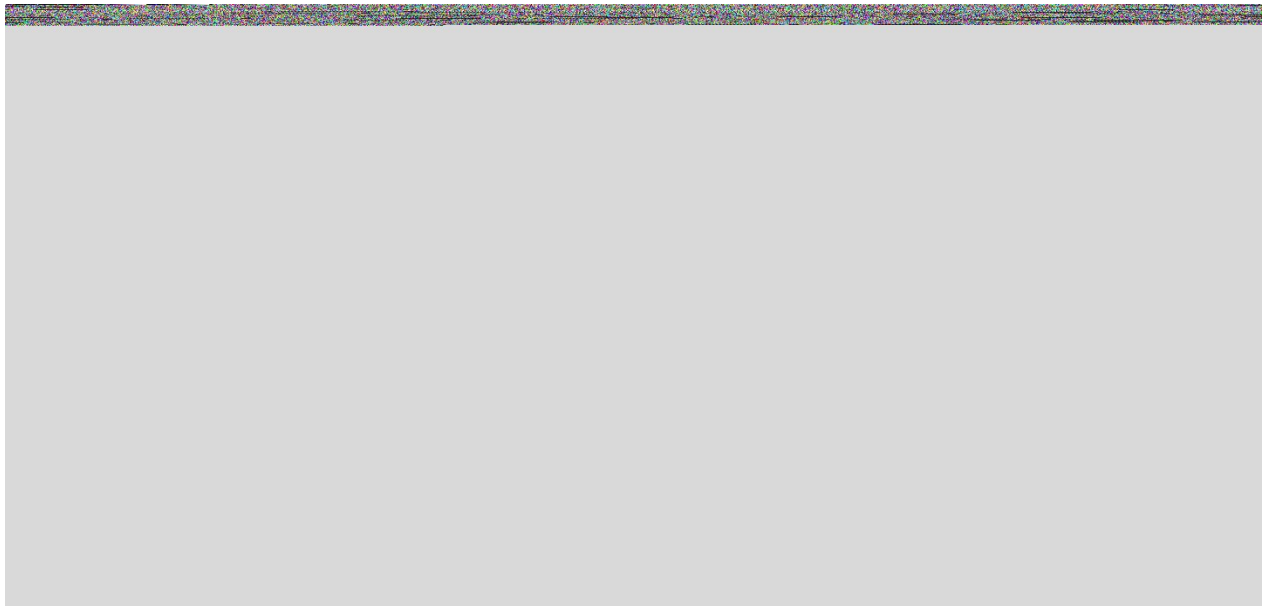


Figura U10. Leitura do arquivo de dados do EIS e preprocessamento dos dados.

Esta célula (“cell”) lê os arquivos de dados do EIS a partir do ‘dataProject.csv’  
Os dados foram nivelados usando `preprocessing.scale(X)`, e uma análise de componentes principais (ACP) ocorreu para decompôr os dados em duas dimensões.



Figura U11. Divisão de dados entre grupos de treinamento e teste.

Dados de EIS são divididos em grupos: 80% treinamento e 20% teste.



Figura U12. O Pyplot do pacote matplotlib foi importado e as figuras são organizadas para aparecerem em uma nova janela.

Ferramentas de plotagem de figuras foram carregadas.





Figura U13. Código de célula para a seleção de núcleos.

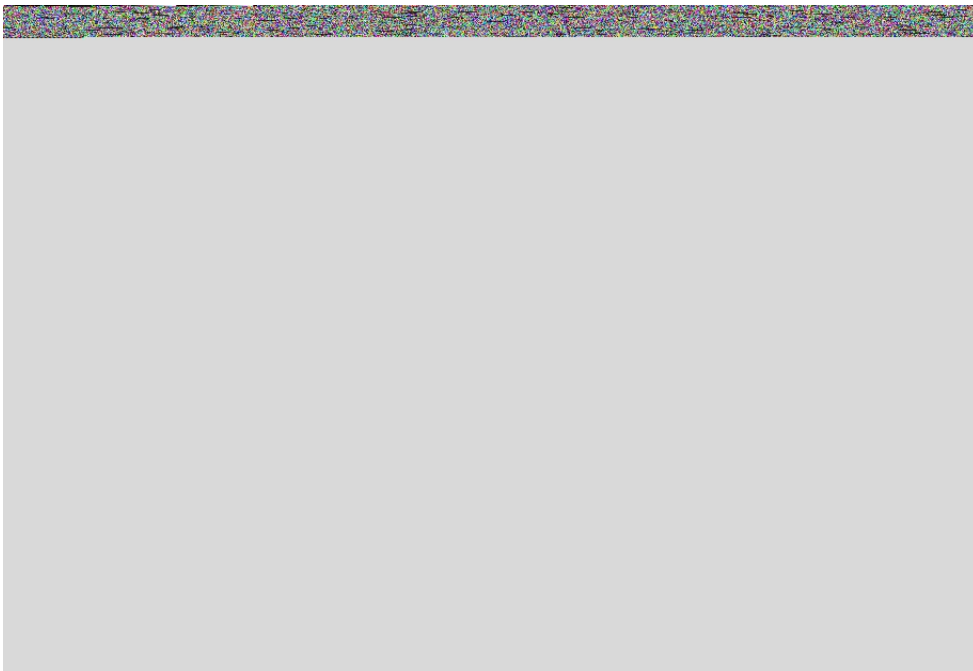


Figura U14. Figura com resultados obtidos com a seleção de diferentes núcleos de códigos.



Figura U15. Validação cruzada e seus resultados.



Figura U16. Encontrando os melhores parâmetros para o núcleo RBF e seus resultados.



Figura U17. Código para visualizar o efeito dos parâmetros no núcleo RBF.

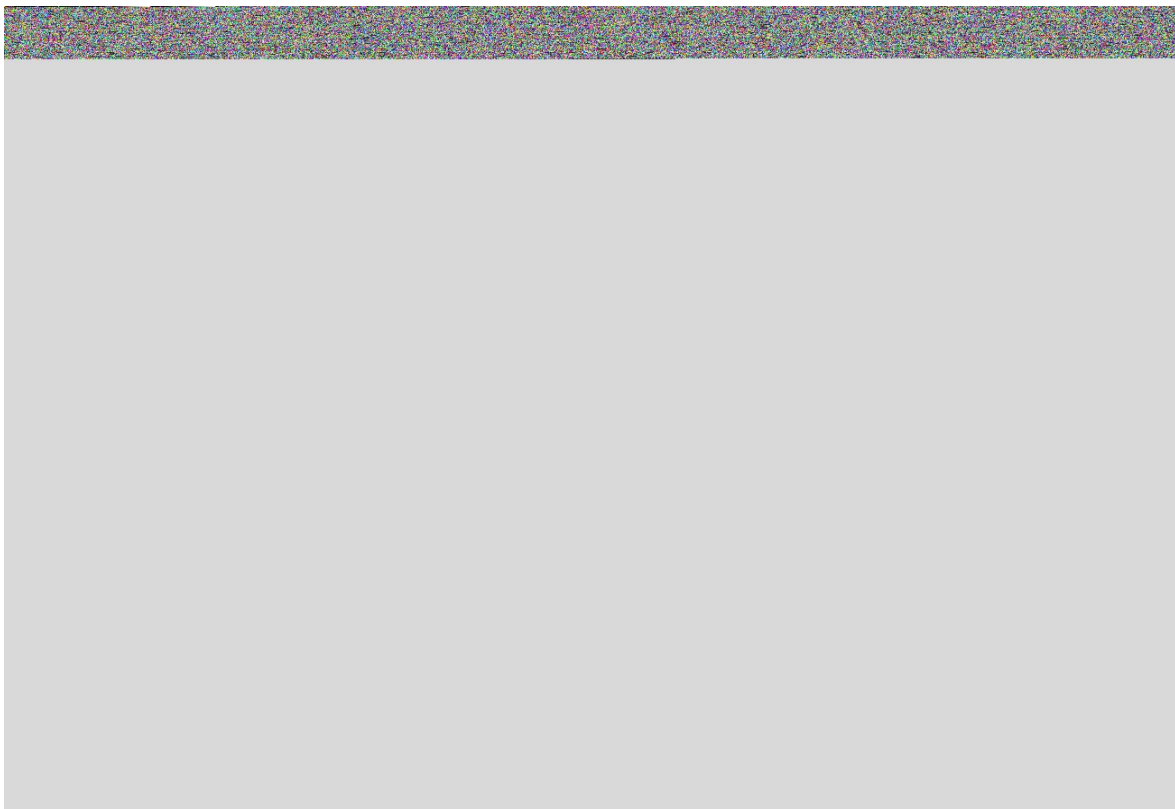


Figura U18. Efeitos dos parâmetros no núcleo RBF.

Se você tiver o seus dados preparados como este, você poderá facilmente modificar e testar o meu código e os códigos disponíveis online.

### **9.3. Scikit-Learn**

<http://scikit-learn.org/stable/index.html>

Se você tiver tempo, verifique este site.

A maioria do meu código e conhecimento é baseado nas informações e exemplos disponíveis deste projeto de fonte aberta.

Boa sorte!

07/01/2018

## Chinese

# SVM Machine Learning for EIS data analysis step-by-step user guide

## 支持向量机机器学习用于电化学阻抗谱数据分析的使用指南

Yue Rong

荣越

UF Biosensor Lab

佛罗里达大学生物传感器实验室

## 10. Software preparation (软件准备)

### 10.1. Download Anaconda installer (下载 anaconda 安装包)

“Anaconda is a freemium open source distribution of the Python and R programming languages for large-scale data processing, predictive analytics, and scientific computing, that aims to simplify package management and deployment.” --[https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution))

“Anaconda 是一个免费开源的 Python 及 R 语言开发环境。它主要用于大规模的数据处理，预测分析和科学计算，致力于简化程序包的管理和使用” —来自维基百科

It includes all the packages and tools we need and is completely free.

该软件包含了我们全部需要的工具

It can be downloaded at:

可以在以下地址下载:

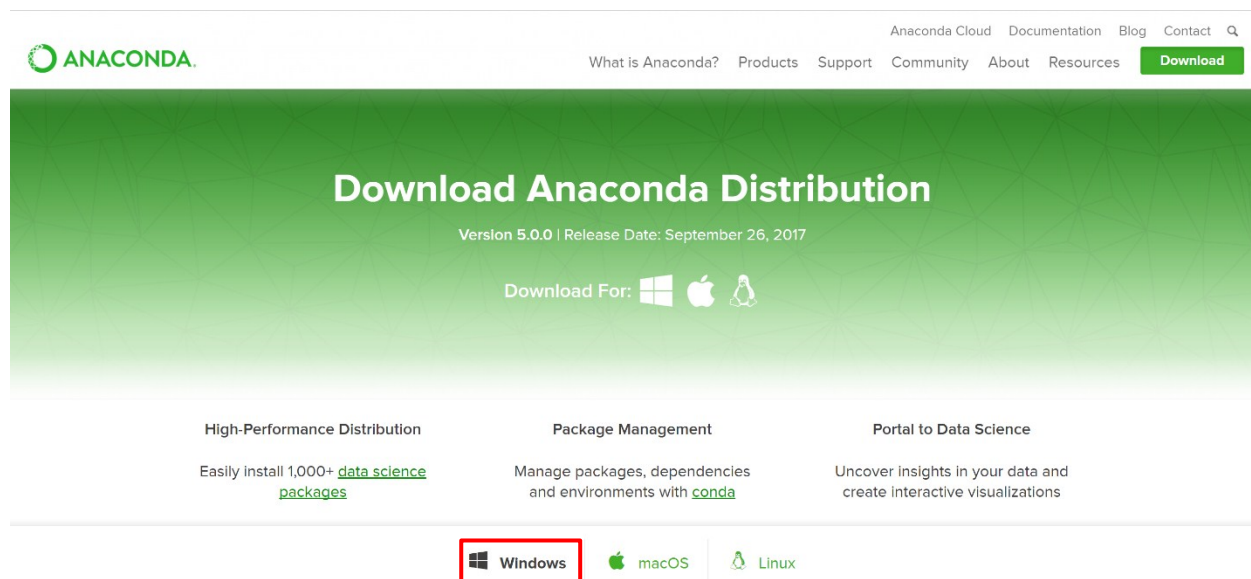


Figure 1 Choose your operating system by clicking “Windows”, “macOS” or “Linux”.

<https://www.continuum.io/downloads>

There are different versions of Anaconda for Windows, macOS and Linux; Python 3.6 and 2.7. It will be about 2 Gb after installation and downloading all packages. We suggest Python 3.6.

该网站提供了 Anaconda 为 Windows, macOS 和 Linux 三种平台的安装包。并且都有 Python 3.6 和 2.7 两个版本。下载安装完成大概需要 2Gb 的硬盘空间。我们建议使用 Python 3.6 版本。

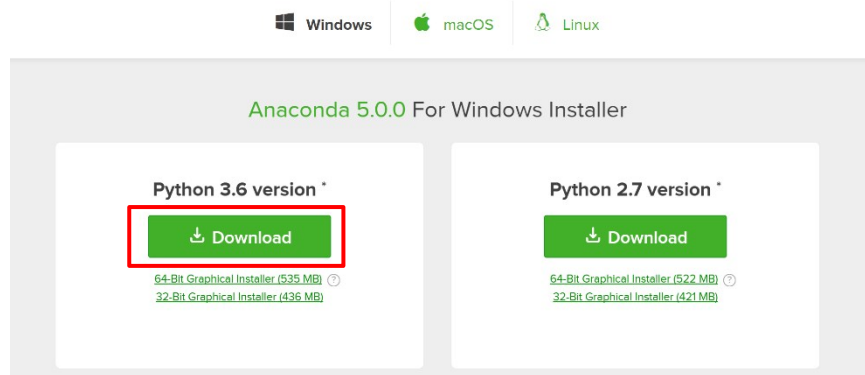


Figure 2 Click to download Python 3.6 version installer (macOS and Linux are similar)

## 10.2. Install Anaconda (安装 Anaconda)

**Double click the installer to launch.** (双击安装包开始)

NOTE: If you encounter any issues during installation, temporarily disable your anti-virus software during install, then re-enable it after the installation concludes. If you have installed for all users, uninstall Anaconda and re-install it for your user only and try again.

注意：如果在安装过程中遇到任何问题，暂时关闭反病毒软件，安装完成后重新开启。如果你选择了“为所有用户”安装，请删除 Anaconda，重新安装并选择“只为自己”安装。

**Click Next.** (点击继续)

**Read the licensing terms and click “I Agree”.** (阅读并点击同意)

Select an install for “Just Me” unless you’re installing for all users (which requires Windows Administrator privileges) and click Next.

选择“只为我”安装，除非你要为所有用户安装（要求 Windows 管理员权限）点击继续。

**Select a destination folder to install Anaconda and click the Next button.** (选择安装位置)

NOTE: Install Anaconda to a directory path that does not contain spaces or unicode characters.

注意：安装目录中不要含有空格或者特殊符号

NOTE: Do not install as Administrator unless admin privileges are required.

注意：不要以管理员身份安装，除非管理员权限需要的时候

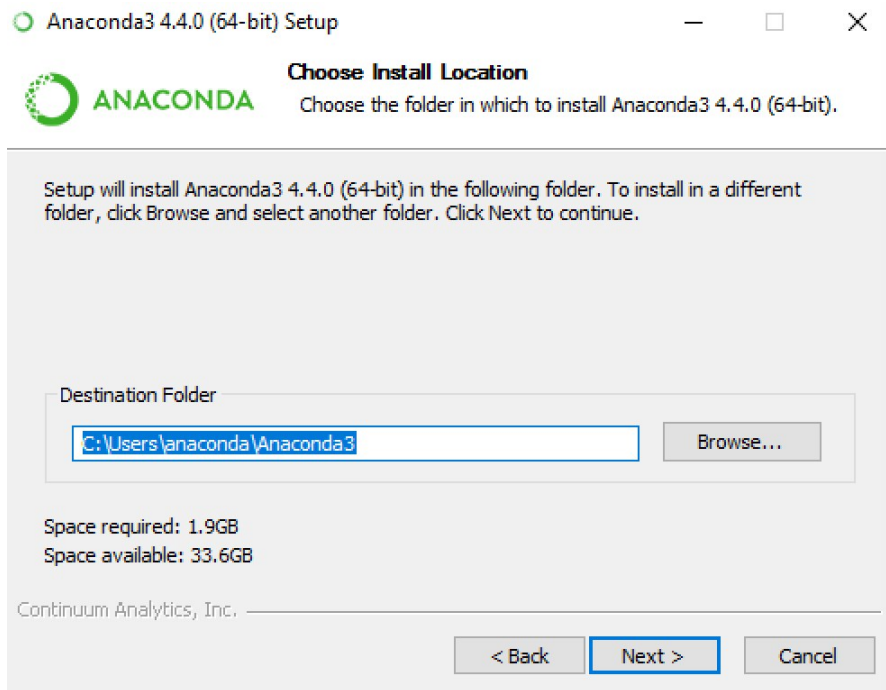


Figure 3 Select a destination folder to install Anaconda and click the Next button

**Choose whether to add Anaconda to your PATH environment variable.**（选择是否将 Anaconda 加入你的 PATH 环节变量）

We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.

我们不推荐把 Anaconda 加入 PATH 环境变量，因为这可能会影响其他软件。所以我们建议从开始菜单打开 Anaconda Navigator 或者 Anaconda Prompt 来使用。

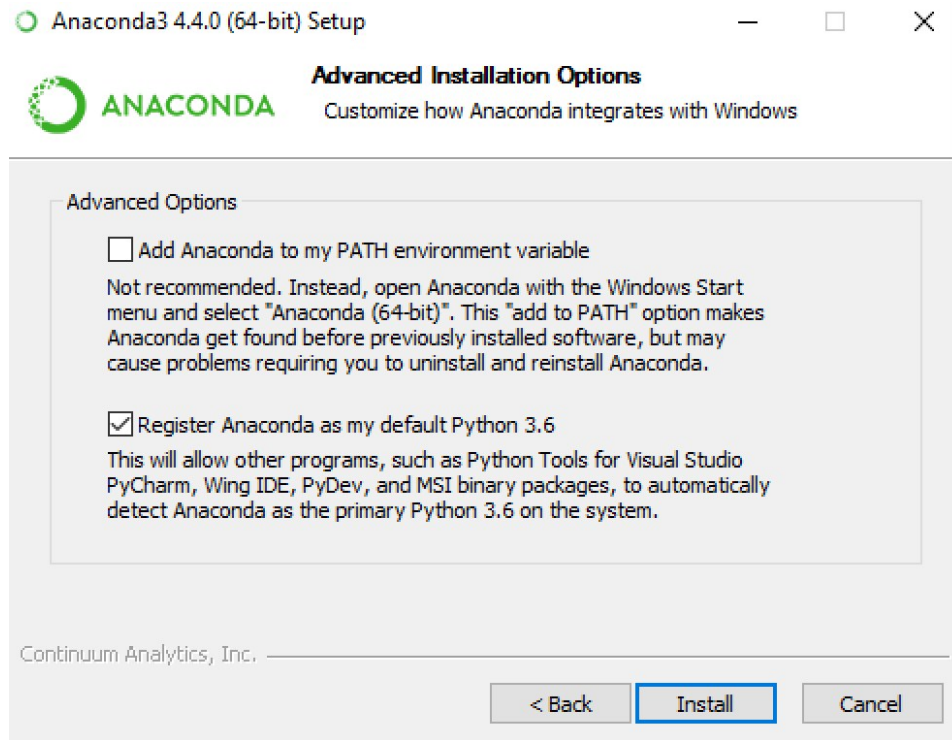


Figure 4 register Anaconda as your default Python 3.6

**Choose whether to register Anaconda as your default Python 3.6.**（选择是否把 Anaconda 设为默认的 Python 3.6）

Unless you plan on installing and running multiple versions of Anaconda, or multiple versions of Python, you should accept the default and leave this box checked.

除非你有多版本的 Anaconda 或者多版本的 Python，你应该接受这个默认设置。

**Click the Install button.**（点击安装）

You can click Show Details if you want to see all the packages Anaconda is installing.

你可以点击产看详情，如果你想看到所有 Anaconda 正在安装的程序包

**Click the Next button.**（点击继续）

After a successful installation, you will see the “Thanks for installing Anaconda” dialog box:

安装完成后，你会看到“感谢安装 Anaconda”的对话框



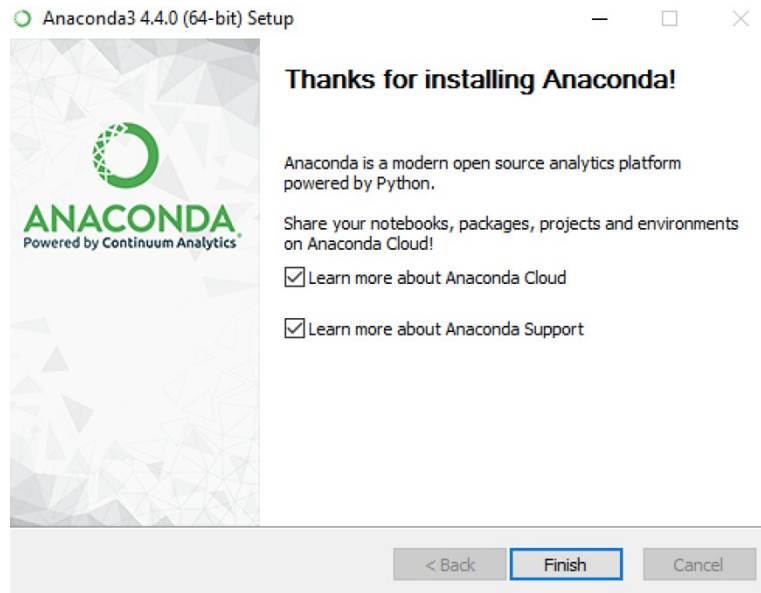


Figure 5 successful installation dialog

**Click the Finish button. (点击完成)**

**After your install is complete, verify it by opening Anaconda Navigator**, a program that is included with Anaconda: from your Windows Start menu, select the shortcut Anaconda Navigator. If Navigator opens, you have successfully installed Anaconda.

安装完成后，通过打开 **Anaconda Navigator** 来确认。如果你能打开 Anaconda Navigator，说明你已经成功安装了 Anaconda。

If you had any issue installing the Anaconda, please visit <https://docs.anaconda.com/anaconda/install/> for help.

如果你在安装过程中遇到其他问题，请访问：<https://docs.anaconda.com/anaconda/install/>

## 11. Getting started (准备开始)

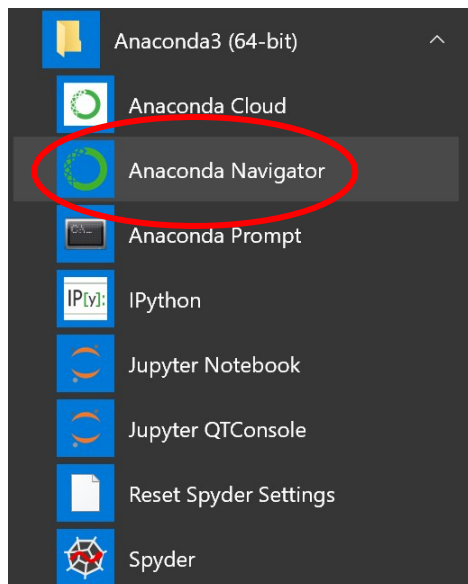
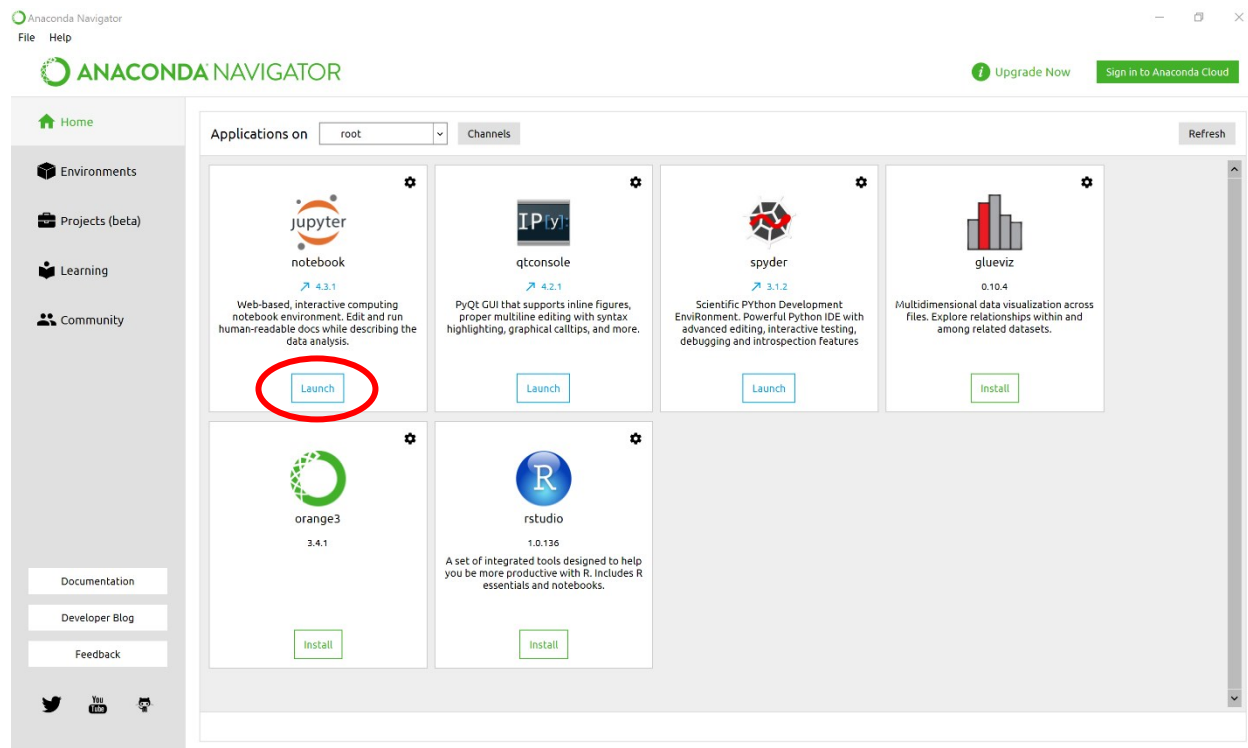


Figure 6 Find Anaconda 3 and open “Anaconda Navigator”

### 11.1. Open Anaconda Navigator (打开 Anaconda Navigator)

Find Anaconda 3 in “Windows Start Menu” and click “Anaconda Navigator” to open  
在开始菜单中找到 Anaconda 3 并双击 “Anaconda Navigator” 打开



### 11.2. Click “Launch” to open “Jupyter Notebook” (点击 “Jupyter notebook” 下的 “Launch” 按钮)

It will open in your internet browser (for me it’s a Chrome) and the default file location is under your system user folder. (for example C:\Users\Username)

Figure 7 Lunch “Jupyter Notebook” in Anaconda Navigator

它会在你默认的网络浏览器（我用的是 Chrome）中打开，默认的文件位置是在你的系统用户文件夹。

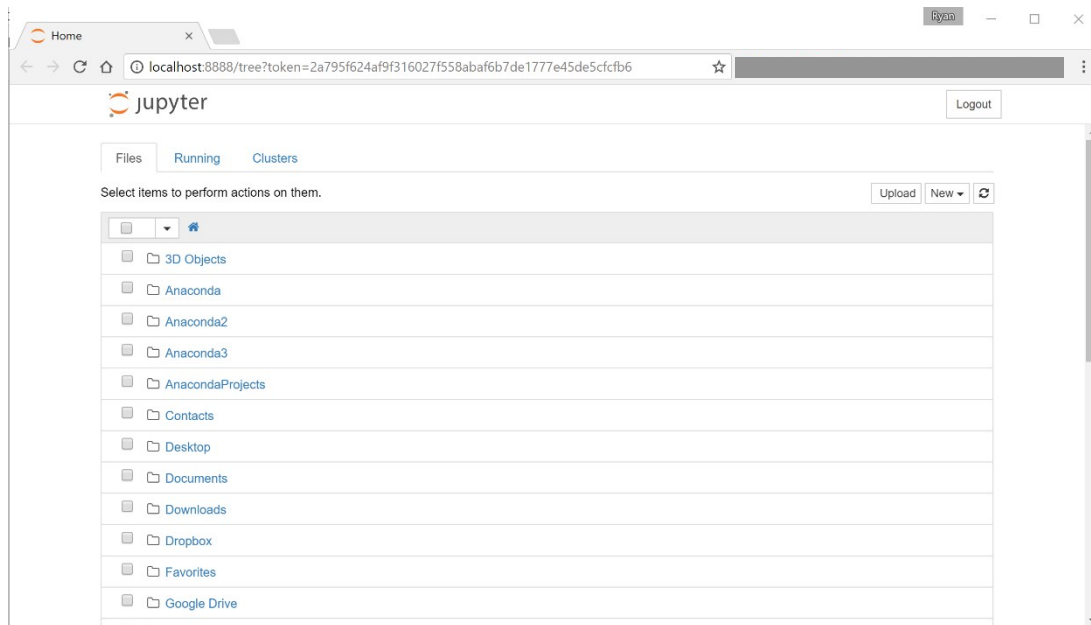


Figure 8 “Jupyter Notebook” dashboard

### 11.3. Run an example code（运行一个测试代码）

Download an example code from the link below and save it to “downloads” folder.

从以下地址下载一个测试代码，并保存到“下载”目录

[http://scikit-learn.org/stable/downloads/plot\\_svm\\_nonlinear.ipynb](http://scikit-learn.org/stable/downloads/plot_svm_nonlinear.ipynb)

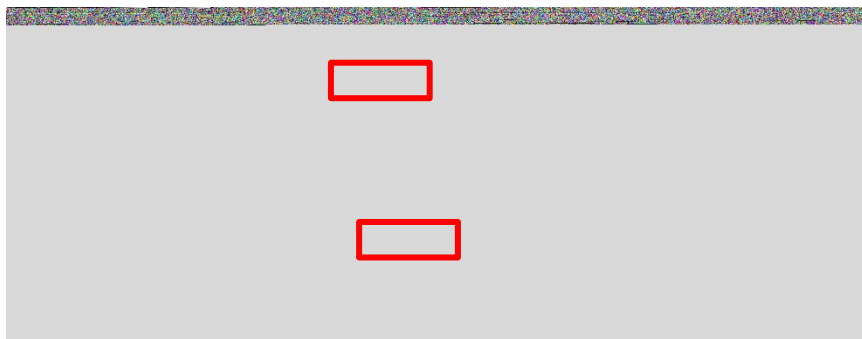
use the file browser to navigate and open the download source code above

通过网页内的文件浏览器找到并打开刚下载测试代码



*Figure 9 Opened example code in “Jupyter Notebook”*

It should look like the screenshot above, and in “Cell” click “Run All” or run cells by clicking the .

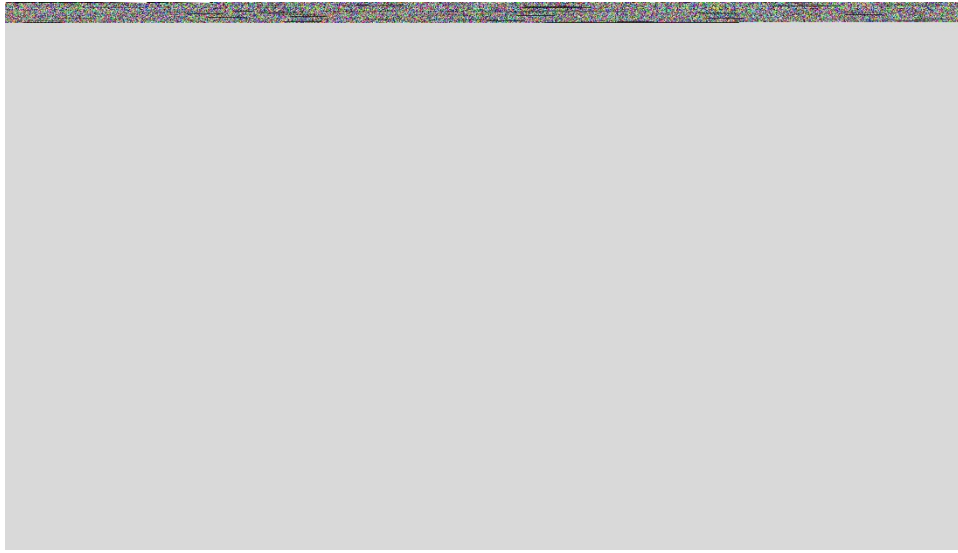


*Figure 10 Run All cells to test the example code*

打开后应该如上图的截屏一样。在“Cell”里点击“Run All”或者单个段落运行。

If it generated a figure like this, your software is good to go. Cheers!

如果它生成了下图，那么你的软件就没问题了！



*Figure 11 Expected output from example code*

## 12. EIS data analysis with SVM machine learning (用支持向量机机器学习对电化学阻抗谱数据进行分析)

### 12.1. Data preparation (数据准备)

Raw EIS data usually includes frequency with real impedance ( $Z'$ ) and imaginary impedance ( $Z''$ ) at that frequency.

电化学阻抗谱的原始数据通常由在特定频率的实阻抗和虚阻抗构成。

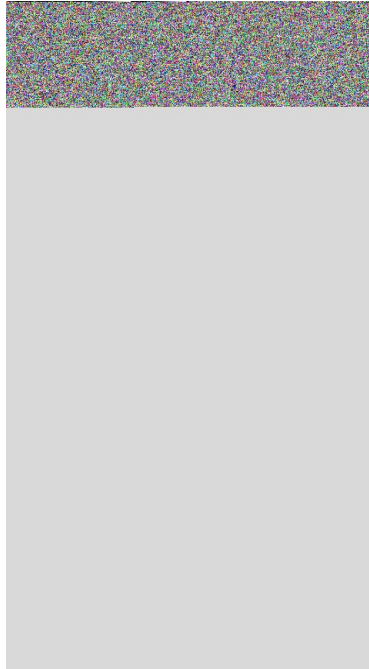


Figure 12 A part of representative raw EIS data.

Make sure all you EIS test are using the same frequency range then **crate an excel sheet** as shown below, where first column are names, second column are labels, each following column is one feature which is either real impedance or imaginary impedance at a frequency.

确定你所有电化学阻抗谱的测试使用的都是相同的频率分布，并按下图建一个新的 excel 表格。



Figure 13 Excel template first step

第一列是名称，第二列是标签，之后的每一列是实阻抗护着虚阻抗。

Then, copy and transpose paste all real impedance (column B in Figure 12) to row 2 in Figure 13 starting at column C. (you may need to transpose paste the data in its original sheet then copy and paste to the template)

然后，把所有图 12 中 B 列的实阻抗数据，转置粘贴到图 13 中第二行，从第 C 列开始的位置（在原表格中先转置再复制粘贴可能更容易）

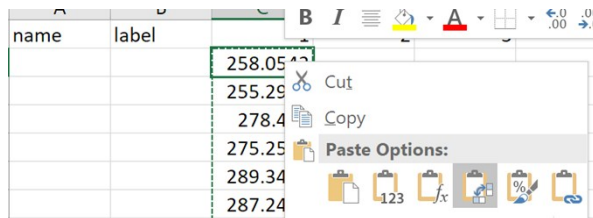


Figure 14 Using transpose paste option to paste all real impedance data in a row.

Then using similar method to copy all imaginary impedance (column C in Figure 12) to row 2 in Figure 13 starting at column CA (I have 76 frequencies in one EIS, I named row 1 from C as 1,2,3...76,77, to 152). Use your own naming system to give each EIS scan a unique name and put it in column A. Set a label (0 or 1 for binary classification) and put it in column B.

然后，用类似的方法，把所有虚阻抗的数据（图 12 中 C 列的数据）转置粘贴到图 13 中，第二行，CA 列开始的位置（在我的电化学阻抗谱中用了 76 个频率，所以第一行从 C 开始命名为 1,2,3 到 76，然后 77,78 一直到 152）。你可以使用你自己的命名系统，在 A 列中给每一个数据列命名，然后在 B 列中给每一个数据设置标签（二进制分类中就是 0 或者 1）



Figure 15 Pasted imaginary impedance into row 2 from column CA to EX (columns D-BY and CC-EV are hidden). Name and label added

Transfer all you EIS data that you are interested in classification (usually with similar experimental setting and a single variable) to the template and save it as \*.csv file

把那些你所有想要分类的的电化学阻抗谱数据（通常是相似的单变量实验）都用同样的方式粘贴到这个模板中，并另存为 csv 格式。

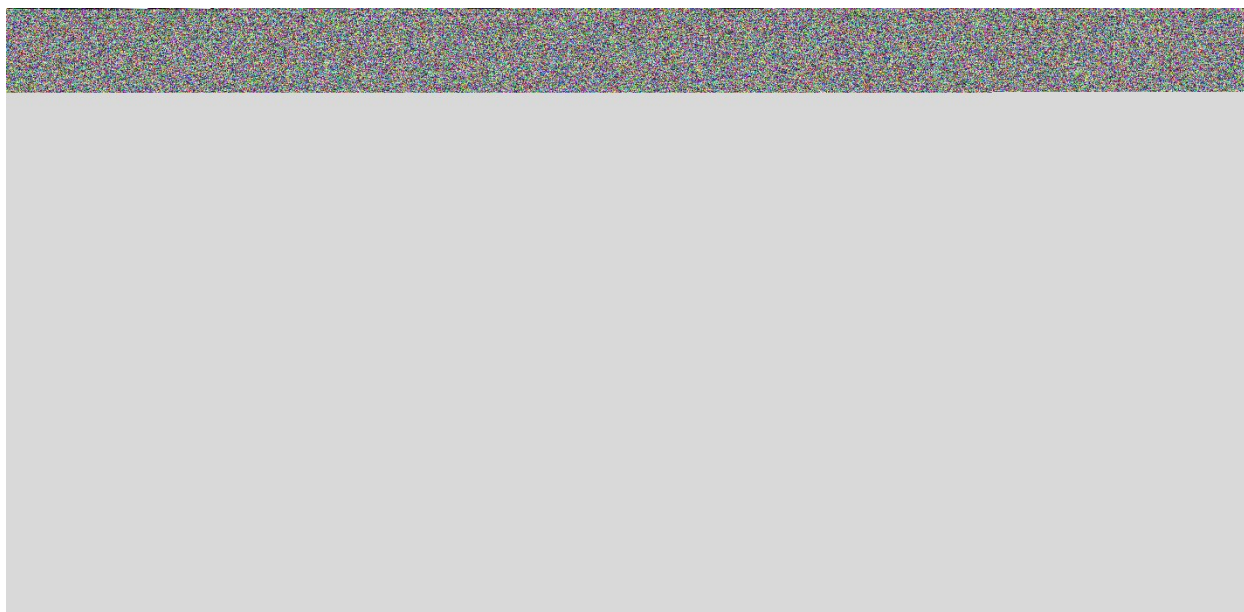


Figure 16 Example raw EIS dataset ready for SVM machine learning

This example file can be downloaded from:

我的实例文件可以从以下地址下载（国内可能需要翻墙）

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>

Each row is a data from one run of a sample (one EIS/CV scan)

The label column is used for supervised classification/regression. It could be binary or quantitate depends on your applications.

每一行代表一个完整的电化学阻抗谱。标签列用于有监督的分类或者回归。它可以是二进制的也可以是定量的。

*For example:*

*In this dataset, "0" represent the blank/negative EIS results with no acetone in the samples; "1" represent the positive results with 5mM acetone concentration in the samples.*

比如说：

在这个数据中，0 代表了空白对照，1 代表了有 5mM 丙酮的样本。

*If you are testing glucose with 0, 1, 10 and 100 mM, you may label them as either "0,1,2,3" or "0,1,10,100" depending on doing classification or regression.*

如果你要检测葡萄糖在 0,1,10 和 100mM 这几个浓度，那你可以给对应数据加上“0,1,2,3”或者“0,1,10,100”的标签。取决于你是要分类分析还是回归分析。

Each column followed is called a “feature” that in EIS is the real or imaginary impedance at a certain frequency.

之后每一列都是一个机器学习中的“特征”，它们都是电化学阻抗谱中某一频率下的实阻抗或者虚阻抗

## 12.2. SVM code for EIS data（支持向量机用于电化学阻抗谱的代码）

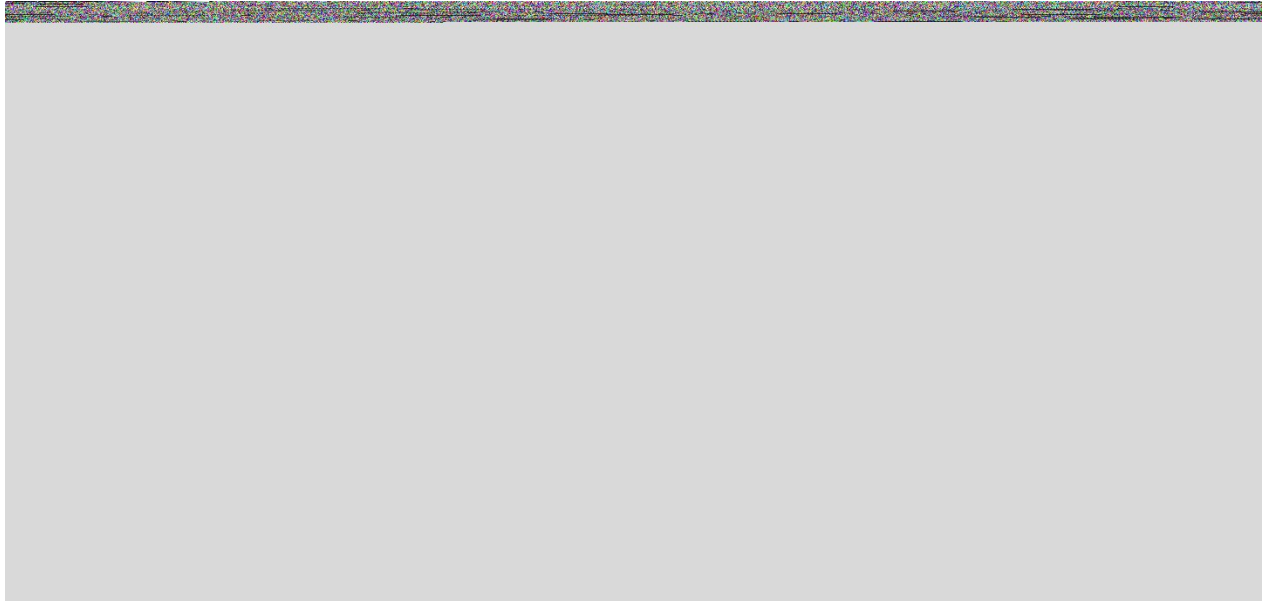
The code can be downloaded from:

代码可以从以下地址下载（国内可能需要翻墙）

<https://www.dropbox.com/s/7ftbm75mpyj6mc1/dataProject.csv?dl=0>



Opened it in “Jupyter notebook” and put the datafile prepared as above in the same folder.  
Run the code by cells.  
用 “Jupyter notebook” 打开，并把之前的数据文件放到相同的目录。



*Figure 17 reading EIS dataset and data preprocessing*

This cell read the EIS dataset from ‘dataProject.csv’

The data was scaled using `preprocessing.scale(X)`. And a principle component analysis (PCA) was performed to decompose the data into two dimensional.

这一个段落作用是从 “dataProject.csv” 读取数据。数据之后通过 `preprocessing.scale(X)` 标准化，通过主成分分析实现降维。



*Figure 18 Split data into training and testing sets.*

EIS data were split into 80% training and 20% testing.

电化学阻抗谱数据组被分为 80% 的训练组和 20% 的检验组



*Figure 19 pyplot from matplotlib package was imported and figures are set to appear in a new window.*

Figures plotting tools was loaded. 制图工具载入



Figure 20 result figure from the kernel selection code 核选择代码的结果图



Figure 21 code cell for kernel selection 核选择的代码



Figure 22 Finding the best parameters for RBF kernel and its results 径向基函数核最佳参数选择的代码核



Figure 23 Cross validation and its result. 交叉验证的代码核结果



Figure 24 Effect of parameters for RBF kernel. 径向基函数核参数改变的效果



Figure 25 Code to visualize effect of parameters for RBF kernel. 径向基函数核参数效果展示的代码

If you have your dataset prepared like this you may easily modify my code or the codes available online and test it out.

如果你的数据用这种方式预处理好了，那么你可以很容易的通过修改我的代码或者使用网上可用的代码来进行测试

### 12.3. Scikit-Learn

<http://scikit-learn.org/stable/index.html>

If you have time, checkout this site.

Most of my code and knowledge is based on the information and example from this open source project.

如果你有时间，请查阅以上网址，我的所有代码都是基于 Scikit-learn 这个开源项目。

Good Luck! 祝你好运!

10/23/2017



