Contenido

[SVM 1](#_Toc509256059)

[REDES NEURONALES 7](#_Toc509256060)

[Keras 7](#_Toc509256061)

# SVM

Se trabajara con los datos de clasificación de vinos de la clase de AED, para predecir la clase de vino según las demás variables

1. Importar las librerías necesarias

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from itertools import cycle

from sklearn import svm, datasets

from sklearn import preprocessing

from sklearn.metrics import roc\_curve, auc, accuracy\_score, average\_precision\_score, precision\_recall\_curve

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import label\_binarize

from sklearn.multiclass import OneVsRestClassifier

from scipy import interp

# Import evaluation metrics Python

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.metrics import roc\_curve, auc, roc\_auc\_score

1. Importar el conjunto de datos

data = pd.read\_csv("churn.csv", sep=",")

data.head(5)

1. Separar las variables X y Y

*y = data["churn"]*

*X = data.drop(["churn"], axis=1)*

*X = X.drop(["phone number"], axis=1)*

*X = X.drop(["state"], axis=1)*

*X = X.drop(["area code"], axis=1)*

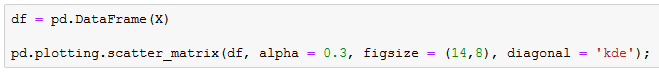
*X['international plan'] = np.where(X['international plan'] == 'no', 0, 1)*

*X['voice mail plan'] = np.where(X['voice mail plan'] == 'no', 0, 1)}*

1. Normalizacion de datos



1. Graficar las variables luego de la transformacion



Que correlaciones observa?

1. Generar un dataset para entrenamiento y otra para test

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.3,

random\_state=123,

stratify=y)

1. Crear el modelo de SVM con un kernel de separación lineal, y ajustar el modelo con los datos de entrenamiento



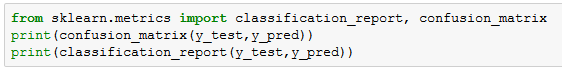
1. Explique que hace la siguiente instruccion:

model.support\_vectors\_

1. Guarde las predicciones y puntuaciones de predicción con los datos de prueba



1. Genere la matriz de confusión para este modelo lineal



Indique su analisis de la tabla generada (recall=sensitivity o sensibilidad) (f1 score:: ver <https://en.wikipedia.org/wiki/F1_score> )

1. Genere la gráfica ROC para el modelo

fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred)

roc\_auc = auc(fpr, tpr)

plt.figure()

plt.plot(fpr, tpr, color='darkorange', lw=1, label='ROC curve (area = %0.2f)' % roc\_auc)

plt.plot([0, 1], [0, 1], color='navy', lw=1, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.show()

1. En el punto 7, se generó el modelo con un parámetro de kernel lineal, repita el proceso variando el kernel para el SVM (polinomial, sigmoide y rbf), represente los resultados mediante una tabla e indique su análisis comparativo

model = svm.SVC(kernel='poly')

model = svm.SVC(kernel='rbf')

model = svm.SVC(kernel='sigmoid')