Machine Learning

**תרגיל 7 – Support Vector Machines (SVMs) and python**

במשימה זו נשתמש בSVM על מנת לקטלג הודעות email כ- spam או non-spam.

email מיוצג בעזרת מספר מאפיינים כמו שכיחות הופעה של מילות מפתח, אורך מילים בcapital letters וכדומה. נשתמש בdata set המכיל 4601 דוגמאות.

מידע על הdata set והפורמט בו הוא שמור אפשר למצוא כאן: <https://archive.ics.uci.edu/ml/datasets/Spambase>

עליכם לבחור רנדומלית 70% מהdata כדוגמאות אימון (training) והשאר ישמש לבדיקות (test).

העזרו בקוד מתרגיל הכיתה הקודם על מנת ללמוד את המודל ולנסות אותו.

עליכם להגיש דו"ח המתייחס לאחוזי דיוק ופונקציות kernel שונות, בקובץ pdf, המכיל את המידע הבא:

תוצאות הניסוי

דווחו על תוצאות הקיטלוג עבור דוגמאות האימון ועבור דוגמאות הבדיקה בשימוש בכל אחת מפונקציות הkernal הבאות:

(a) Linear

(b) Quadratic

(c) RBF

עבור כל פונקצית kernel עליכם לנסות ערכי regularization parameter שונים (הפרמטר C) ולדווח על הערך הטוב ביותר. ערך ה-C הטוב ביותר הוא זה שעבורו התקבלו התוצאות הטובות ביותר על קבוצת הבדיקה.

כיתבו את הדוח בצורת טבלת השוואה.

הגישו לתיבת ההגשה במודל.

# import libraries  
import pandas as pd  
import numpy as np  
import seaborn as sns  
import matplotlib.pyplot as plt  
%matplotlib inline  
# Importing Data file  
data = pd.read\_csv('bc2.csv')  
dataset = pd.DataFrame(data)  
dataset.columns  
dataset.info()  
dataset.describe().transpose()  
dataset = dataset.replace('?', np.nan)  
dataset = dataset.apply(lambda x: x.fillna(x.median()),axis=0)  
  
# converting the hp column from object 'Bare Nuclei'/ string type to float  
dataset['Bare Nuclei'] = dataset['Bare Nuclei'].astype('float64')  
dataset.isnull().sum()  
  
from sklearn.model\_selection import train\_test\_split  
  
# To calculate the accuracy score of the model  
from sklearn.metrics import accuracy\_score, confusion\_matrix  
  
target = dataset["Class"]  
features = dataset.drop(["ID","Class"], axis=1)  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(features,target, test\_size = 0.2, random\_state = 10)  
from sklearn.svm import SVC  
  
# Building a Support Vector Machine on train data  
svc\_model = SVC(C= .1, kernel='linear', gamma= 1)  
svc\_model.fit(X\_train, y\_train)  
  
prediction = svc\_model .predict(X\_test)  
# check the accuracy on the training set  
print(svc\_model.score(X\_train, y\_train))  
print(svc\_model.score(X\_test, y\_test))