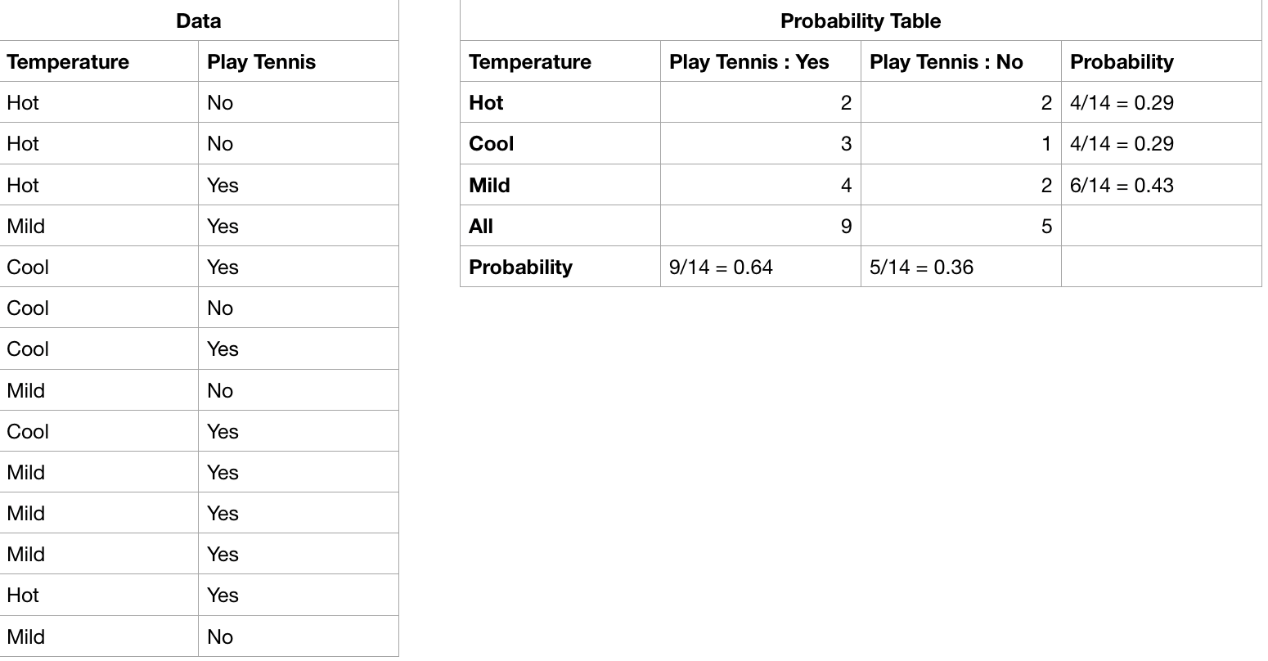
Machine Learning

**תרגיל 2 – Naïve Bayes**

נתון מידע עבור שחקן הטניס ג'ו:

[](http://daydreamingnumbers.com/wp-content/uploads/2017/10/Data-Example.png)  
אנחנו רוצים לבדוק האם הטענה שכאשר הטמפרטורה בינונית (mild) ג'ו ישחק טניס תקפה. אנחנו זקוקים להסתברות שג'ו ישחק טניס בהינתן טמפרטורה בינונית, כלומר:

P(Joe Plays | Mild Temperature)

לפי הכלל של Bayes,

P(Joe Plays | Mild Temperature)= P(Mild Temperature | Joe plays) P(Joe Plays) / P(Mild Temperature)

(4/9) \* (0.64) / (0.43) = 0.65

כלומר, כאשר הטמפרטורה בינונית, יש סיכוי טוב (גבוה מ 0.5) שג'ו ישחק טניס.

נרחיב את הדוגמא הזאת ונבנה את מודל הnaive Bayes בpython.

ראשית, נייבא את כל הספריות הנצרכות:

import numpy as np

import pandas as pd

from sklearn.naive\_bayes import GaussianNB

from sklearn.preprocessing import LabelEncoder

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

from sklearn.metrics import accuracy\_score

עכשיו, נטען את קובץ הנתונים בעזרת המתודה read\_csv מתוך pandas

play\_tennis = pd.read\_csv("PlayTennis.csv")

הכנסו אל הקובץ והתרשמו מתוכנו (נראה מוכר?). העמודה האחרונה מכילה את הקיטלוג. הציגו את תחילת הקובץ.

play\_tennis.head()

מה הוצג למסך?

Sklearn דורש שימוש במאפיינים שהם מערכים מספריים. עלינו להמיר את הערכים בטבלה לערכים מספריים.

LabelEncoder ממיר ערכים קטגוריים למספרים בטווח 0-(n-1), כאשר n הוא מספר הערכים האפשרי עבור המשתנה. לדוגמא, עבור Outlook, קיימים 3 ערכים קטגוריים: Overcast, Rain, Sunny. הם יומרו לערכים 0,1,2 לפי סדר אלף-ביתי.

number = LabelEncoder()

play\_tennis['Outlook'] = number.fit\_transform(play\_tennis['Outlook'])

play\_tennis['Temperature'] = number.fit\_transform(play\_tennis['Temperature'])

play\_tennis['Humidity'] = number.fit\_transform(play\_tennis['Humidity'])

play\_tennis['Wind'] = number.fit\_transform(play\_tennis['Wind'])

play\_tennis['Play Tennis'] = number.fit\_transform(play\_tennis['Play Tennis'])

איך נראית הטבלה עכשיו?

כעת אנחנו מוכנים ליצירת המודל. נגדיר את משתני המאפיינים והמטרה (הקיטלוג הסופי).

features = ["Outlook", "Temperature", "Humidity", "Wind"]

target = "Play Tennis"

ניצור חלוקה לדוגמאות אימון, training, ודוגמאות בדיקה, test.

נבנה את המודל בעזרת דוגמאות האימון, ונאמת אותו בעזרת דוגמאות הבדיקה.

features\_train, features\_test, target\_train, target\_test = train\_test\_split(play\_tennis[features],

play\_tennis[target],

test\_size = 0.33,

random\_state = 54)

עתה נבנה את המודל עצמו.

model = GaussianNB()

model.fit(features\_train, target\_train)

מה לדעתכם עושה המתודה fit?

כעת אנחנו מוכנים לנסות את המודל על דוגמאות הבדיקה בעזרת מדד הדיוק accuracy, המונה את מספר הדוגמאות שהמודל קיטלג נכון.

pred = model.predict(features\_test)

accuracy = accuracy\_score(target\_test, pred)

מהי רמת הדיוק של המודל ביחס לדוגמאות הבדיקה?

כעת, אנחנו רוצים לחזות את הסיכוי לשחק טניס ביום מסוים עם המאפיינים הבאים:

|  |  |  |  |
| --- | --- | --- | --- |
| **Outlook** | **Temperature** | **Humidity** | **Wind** |
| Rain | Mild | High | Weak |

print model.predict([[1,2,0,1]])

מה חזה המודל? האם ג'ו ישחק ביום כזה או לא?

לשם בדיקת נכונות, מידת הדיוק של המודל הנלמד היא 0.80000000000000004, וג'ו יקבל חיזוי של 1, כלומר, ישחק טניס ביום הזה.