תרגיל 4

הערות להגשה:

- יש להשתמש בפייטון ולממש את ה- classifiers מ-scratch . לדוגמא, לא להשתמש בשום ספריה של machine learning. כמובן שאפשר להשתמש ב-numpy ולהתבסס על השקפים בשיעור .
- עבור כל תרגיל יש להגיש את הקוד בפייטון + צילומי מסך של התוצאות. **שימו לב**- תרגילים ללא צילומי מסך לא יבדקו!

Spark + Naïve Bayes:

ראינו בשיעור classifier של Naïve Bayes המבוסס על ספירה של המופעים של מילים ביחס למס' ההודעות הקיים במילון .

1. שנו את הקוד וצרו classifier המבוסס על ספירה של מילים ביחס למס' **המילים** הקיימות במילון

2. Laplace's smoothing הוסיפו מנגנון החלקה

Linear and logistic regression:

מצ"ב רשימת נתונים על מחירי דירות בעיר אחת בחו"ל:
הנתונים הם:

1מחירי המכירה המקומיים, במאות דולרים;
2, מספר חדרי האמבטיה;
3, שטח האתר באלפי מטרים רבועים;
4 גודלו של שטח המחיה באלפי מטרים רבועים;
5 מספר המוסכים;
6, מספר החדרים;
7, מספר חדרי שינה;
8, הגיל בשנים;
9, סוג הבנייה
10, סוג אדריכלות
11, האם יש מכבי אש (0 אין 1 יש).
12.מחיר המכירה

קראו את הנתונים ב- python ו:

1. בנו מודל **linear** regression בפייטון אשר חוזה את מחיר הדירה על פי הנתונים:location,bedrooms,sqft.

השתמשו ב-70% מהנתונים עבור בניית המודל, ב-30% הנותרים בידקו את המודל שיצרתם .

2. דווחו את התוצאות ב-**mean squared-error .** יש להשוות את התוצאות של המודל לנתונים הקיימים .
3. הוסיפו את הפיצ'רים:status, title ובדקו האם ה- **mean squared-error** קטן יותר מסעיף קודם

4. בנו **logistic** regression classifier בפייטון אשר חוזה האם secured=1 או secured=9 השתמשו באותם פיצ'רים מהשאלה הקודמת+price . השתמשו ב-70% מהנתונים עבור בניית המודל, ב-30% הנותרים בידקו את המודל שיצרתם .
5. דווחו את התוצאות במונחים של **accuracy, recall, precision and F-measure**

דרך קלה יותר לקרוא קובץ xl לפי עמודות בעזרת ספרית pandas:

```
f test_find_coefficients_multival():
  df_train = pd.read_csv('../../../Downloads/house_prices_train.csv')
  df_train = df_train.sample(frac=1).reset_index(drop=True)

  x = df_train[['OverallQual', 'GrLivArea', 'GarageCars']][:1000]
  y = df_train['SalePrice'][:1000]

  x_test = df_train[['OverallQual', 'GrLivArea', 'GarageCars']][1000:-1].reset_index(drop=True)
  y_test = df_train['SalePrice'][1000:-1].reset_index(drop=True)
```