DS 5220: Supervised Machine Learning and Learning Theory (Spring 2023) Dr. Roi Yehoshua
Student name: (Due) March 13, 2023

PS2: Classification

1 Classification Metrics (10%)

A new COVID-19 test-kit has been released and you need to assess its effectiveness. Results of the experiments performed with the kit are summarized in the following confusion matrix:

		Predictions (by the test)	
		+	-
Actual (covid status)	+	33	1
	-	3	72

1. How many false positives are there? how many false negatives?
2. What is the accuracy of the test?
3. What are the precision and recall of the test?
4. What is the F_1 score?
5. Would you rather have a higher precision or a higher recall in this case? Explain.

2 Logistic Regression (20%)

In this exercise you will implement a logistic regression model to predict whether a student gets admitted into a university based on historical data from previous applicants.

1. Download the notebook `LogisticRegressionEx.ipynb` and the text file `exams_data.txt` from Canvas.
2. Follow the instructions in the notebook and complete the code for the functions written there.

3 k-Nearest Neighbors (KNN) (15%)

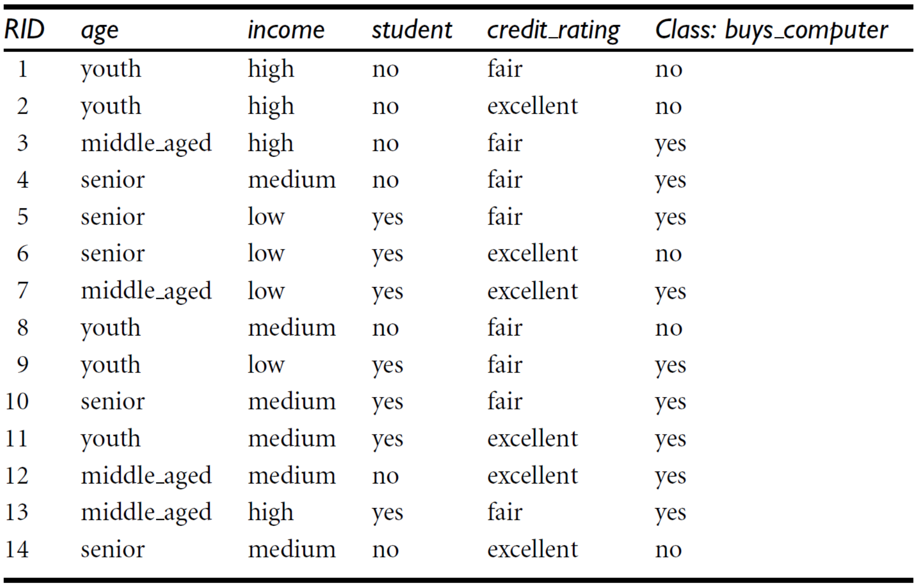
Consider the following data set comprised of three numerical features (f_1, f_2, and f_3) and one binary output y:

Example	f_1	f_2	f_3	y
$\mathbf{x}_1$	1	4	1	1
$\mathbf{x}_2$	1	2	3	1
$\mathbf{x}_3$	0	0	1	1
$\mathbf{x}_4$	-1	4	0	1
$\mathbf{x}_5$	1	0	-2	0
$\mathbf{x}_6$	-1	-1	1	0
$\mathbf{x}_7$	0	-4	0	0
$\mathbf{x}_8$	1	0	-3	0

Based on this data set, classify a new vector $\mathbf{x} = (1, 0, 1)$ using KNN with $k = 3$ and Manhattan distance. Show your work.

4 Naive Bayes Classification (15%)

The following table shows a part of a customer database of an electronic store:



RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Use Naive Bayes to predict whether the following customer will buy a computer:

$$\mathbf{x} = (\text{age} = \text{youth, income} = \text{medium, student} = \text{yes, credit\_rating} = \text{fair})$$

Don't use Laplace smoothing. Show your work.

5 Spam Filter (40%)

A spam filter is a computer program that classifies e-mail messages as either spam (unwanted) or ham (wanted). The spam filter maintains a set of features. A feature is a characteristic of messages that the spam filter can use to distinguish ham from spam. A feature could be simply the presence of a word in the message, for example, "the message contains the word 'lottery'", and it could be much more complex/powerful, for example, "the subject line is all capitals" or "the message body mentions many Internet domains".

Given (1) a set of features, (2) a set of messages known to be ham, and (3) a set of messages known to be spam, the spam filter learns how to classify incoming messages as ham or spam.

In this exercise we are going to train a spam filter using the Apache SpamAssassin public mail corpus.

1. Download the files `20030228_easy_ham.tar.bz2` and `20030228_spam.tar.bz2` from `https://spamassassin.apache.org/old/publiccorpus/`. The first file contains 2500 ham messages, and the second one contains 500 spam messages.
2. Unzip the datasets and familiarize yourself with the data format.
3. Use Python's `email` module to parse the email messages (see code example at the end).
4. Apply text preprocessing techniques to convert each email into a feature vector. These may include tokenization of the text into words, removing stop words, replacing all URLs with "URL", TF-IDF vectorization, etc.
5. Split the emails into training and test sets.
6. Try out several classifiers (e.g., Naive Bayes, KNN, logistic regression) and see if you can build a great spam classifier, with both high recall and high precision.
7. Show the confusion matrix and the classification report of your best classifier on both the training and the test set.
8. How does the size of the training set affect the classifier's performance? Train your classifier with different training set sizes, and show the learning curve (test set error vs. training set size).
9. Intuitively, some tokens may be particularly indicative of an email being in a particular class. We can get an informal sense of how indicative token i is for the spam class by looking at:

$$\log\left(\frac{P(\text{token } i|\text{email is spam}}{P(\text{token } i|\text{email is ham}}\right)$$

   Using this measure, find the 10 tokens that are most indicative of the spam class (i.e., have the highest positive value on the measure above).

Example for parsing an email message:

```
import email
from bs4 import BeautifulSoup

def parse_email(file_path):
    with open(file_path, ’rb’) as f:
        msg = email.message_from_bytes(f.read())

    # Get email headers
    subject = msg.get(’Subject’)

    # Read email’s body
    body = str(msg.get_payload())

    # Remove HTML tags
    body = BeautifulSoup(body).get_text()

    return subject, body
```