# Machine Learning In-Class Lab

The goal of this lab is to introduce students to machine learning.

## 1. Install Python libraries

pip install passwordmeter

pip install numpy

pip install scipy

pip install scikit-learn

## 2. Download sample passwords

Check LearningSuite

## 3. Create a Python script to calculate the strength of all the passwords

### a. Import passwordmeter

### b. Import machine learning libraries

from sklearn.feature\_extraction.text import TfidfVectorizer

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

from sklearn.linear\_model import LogisticRegression

### c. Create a list variable for coded passwords

### d. Loop through every password in the password file

### e. Strip off the new line from the password

### f. Get the strength of the password

strength, improvements = passwordmeter.test(password)

### g. Multiply the strength by 100 and turn it into an integer

### h. Add the password and the strength as a nested list to the variable from b

e.g., [07606374520,10]

## 4. Understand TF-IDF

For this analysis we are going to be using: term frequency–inverse document frequency. Please look it up and summarize it in your own words.

Type your answer here.

## 5. Create a function to tokenize the password

Add this function to your Python script

def getTokens(inputString):

tokens = []

for i in inputString:

tokens.append(i)

return tokens

## 6. Get the passwords and strengths ready

### a. Create a variable for the strengths and put them in it

### b. Create a variable for the passwords and put them in it

## 7. Create a vectorizer and vectorize

vectorizer = TfidfVectorizer(tokenizer=getTokens)

X = vectorizer.fit\_transform(passwords)

## 8. Understand training and test sets

In your own words explain why we use training and test sets.

Type your answer here.

## 9. Create the training and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, strengths, test\_size=0.20, random\_state=42)

This creates a test set that is 20% of the data.

## 10. Using a Logistic regression, fit a line to the training data

lgs = LogisticRegression(penalty='l2',multi\_class='ovr')

lgs.fit(X\_train, y\_train)

## 11. Test the test data using the line generated from the logistic regression

print(lgs.score(X\_test, y\_test))

## 12. Run your script

What score did you get back – the percentage of accurate classifications of the test data by your algorithm?

Type your answer here.

## 13. Test some passwords

Create a list of at least 10 passwords

Test them using the following:

X\_predict = vectorizer.transform(passwords\_to\_predict)

y\_predict = lgs.predict(X\_predict)

print(y\_predict)

What were your results (put in the password and the score)?

Type your answer here.

Are these the results that you would expect?

Type your answer here.

Do you think the results are due to the password scoring engine or your algorithm and why? Information about passwordmeter can be found here: <https://github.com/cadithealth/passwordmeter>

Type your answer here.