Documentație Machine Learning

Funcția de calcularea a acurateți:

def accuracy\_score(y\_true, y\_pred):  
 return (y\_true == y\_pred).mean()

Citirea datelor:

# Data reading  
data\_train\_samples = pd.read\_csv('data/train\_samples.txt', sep=" ", header=None)  
data\_train\_labels = pd.read\_csv('data/train\_labels.txt', sep=" ", header=None)  
  
data\_validation\_samples = pd.read\_csv('data/validation\_samples.txt', sep=" ", header=None)  
data\_validation\_labels = pd.read\_csv('data/validation\_labels.txt', sep=" ", header=None)  
  
data\_test\_samples = pd.read\_csv('data/test\_samples.txt', sep=" ", header=None)  
  
train\_ids = data\_train\_samples[0]  
train\_data = data\_train\_samples[1]  
train\_labels = data\_train\_labels[1]  
  
validation\_ids = data\_validation\_samples[0]  
validation\_data = data\_validation\_samples[1]  
validation\_labels = data\_validation\_labels[1]  
  
test\_ids = data\_test\_samples[0]  
test\_data = data\_test\_samples[1]  
  
ytrain = train\_labels.astype('int')  
yvalidation = validation\_labels.astype('int')

Preprocesarea si scalarea textului folosind clasa CountVectorizer:

# Text preprocessing and scaling  
cv = CountVectorizer(encoding='str', strip\_accents='unicode')  
  
xtrain = cv.fit\_transform(train\_data)  
xvalidation = cv.transform(validation\_data)  
xtest = cv.transform(test\_data)  
  
xtrain = xtrain.toarray()  
xvalidation = xvalidation.toarray()  
xtest = xtest.toarray()

**Modelul MultinomialNB:**

alphaList = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.01, 0.02, 0.03,  
 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.001]  
print('--------------')  
for alphaValue in alphaList:  
 mnb = MultinomialNB(alpha=alphaValue)  
 mnb.fit(xtrain, ytrain)  
 predicted = mnb.predict(xvalidation)  
 print(alphaValue, accuracy\_score(predicted, yvalidation), sep=" ")  
  
alphaList = [0.3, 0.31, 0.32, 0.33, 0.34, 0.35, 0.36, 0.37, 0.38, 0.39, 0.4, 0.41, 0.42, 0.43, 0.44, 0.45, 0.46, 0.47,  
 0.48, 0.49, 0.5]  
print('--------------')  
for alphaValue in alphaList:  
 mnb = MultinomialNB(alpha=alphaValue)  
 mnb.fit(xtrain, ytrain)  
 predicted = mnb.predict(xvalidation)  
 print(alphaValue, accuracy\_score(predicted, yvalidation), sep=" ")

Tabelul rezultat este (partea importanta):

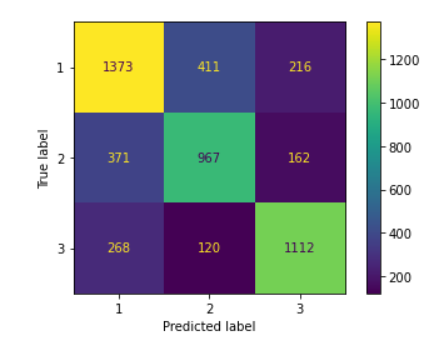
|  |  |
| --- | --- |
| Alpha (Additive smoothing) | score |
| 0.1 | 0.6828 |
| 0.2 | 0.687 |
| 0.3 | 0.6896 |
| 0.31 | 0.6898 |
| 0.32 | 0.69 |
| 0.33 | 0.6904 |
| 0.34 | 0.69 |
| 0.4 | 0.6902 |
| 0.5 | 0.6884 |

Folosindumă de acest tabel am ales alpha = 0.33 si am facut predictiile.

print('--------------')  
mnb = MultinomialNB(alpha=0.33)  
mnb.fit(xtrain, ytrain)  
predicted = mnb.predict(xvalidation)  
print(0.33, accuracy\_score(predicted, yvalidation), sep=" ")  
  
cm = confusion\_matrix(yvalidation, predicted)  
print(cm)

predictedSubmission = mnb.predict(xtest)  
  
output = open('data/test\_labels.txt', 'w')  
output.write('id,label\n')  
for i in range(len(test\_ids)):  
 output.write(str(test\_ids[i]) + ',' + str(predictedSubmission[i]) + '\n')  
output.close()

Matricea de confuzie:



**Modelul Perceptron:**

# Model2  
alphaList = [0.1, 0.01, 0.001, 0.0001, 0.00001, 0.000001]  
print('--------------')  
for alphaValue in alphaList:  
 pr = Perceptron(alpha=alphaValue, penalty='l2') #eta0 = 1  
 pr.fit(xtrain, ytrain)  
 predicted = pr.predict(xvalidation)  
 print(alphaValue, accuracy\_score(predicted, yvalidation), sep=" ")

Tabelul rezultat este:

|  |  |
| --- | --- |
| Alpha(regularizare) | score |
| 0.1 | 0.4014 |
| 0.01 | 0.4098 |
| 0.001 | 0.4642 |
| 0.0001 | 0.574 |
| 0.00001 | 0.536 |
| 0.000001 | 0.6398 |

Folosindumă de acest tabel am ales alpha = 0.000001 și am făcut predicțiile. Learning rate-ul (eta0) a ramas egal cu 1.

pr = Perceptron(alpha=0.000001, penalty='l2') #eta0 = 1  
pr.fit(xtrain, ytrain)  
predicted = pr.predict(xvalidation)  
print(0.000001, accuracy\_score(predicted, yvalidation), sep=" ")  
  
cm = confusion\_matrix(yvalidation, predicted)  
print(cm)  
  
predictedSubmission = pr.predict(xtest)  
output = open('data/test\_labels.txt', 'w')  
output.write('id,label\n')  
for i in range(len(test\_ids)):  
 output.write(str(test\_ids[i]) + ',' + str(predictedSubmission[i]) + '\n')  
output.close()

Matricea de confuzie:

