**Question 1**

"self.tokenize = nltk.tokenize.word\_tokenize"

training accuracy: 0.7075333333333333

validation accuracy: 0.6943333333333334

testing accuracy: 0.70555

"self.tokenize = nltk.tokenize.WhitespaceTokenizer().tokenize "

training accuracy: 0.7212

validation accuracy: 0.7112666666666667

testing accuracy: 0.70925

"self.tokenize = nltk.tokenize.SpaceTokenizer().tokenize"

training accuracy: 0.7211333333333333

validation accuracy: 0.7112666666666667

testing accuracy: 0.7092

(training set: 0.2, validation set: 0.3, testing set: 0.5)

training accuracy: 0.723

validation accuracy: 0.7086

testing accuracy: 0.7102

(training set: 0.3, validation set: 0.3, testing set: 0.4)

training accuracy: 0.7212

validation accuracy: 0.7112666666666667

testing accuracy: 0.70925

(training set: 0.4, validation set: 0.3, testing set: 0.3)

training accuracy: 0.72045

validation accuracy: 0.7080666666666666

testing accuracy: 0.7135333333333334

(training set: 0.5, validation set: 0.3, testing set: 0.2)

training accuracy: 0.71796

validation accuracy: 0.7099333333333333

testing accuracy: 0.7168

(training set: 0.6, validation set: 0.3, testing set: 0.3)

training accuracy: 0.7177333333333333

validation accuracy: 0.7078666666666666

testing accuracy: 0.7262

(training set: 0.3, validation set: 0.2, testing set: 0.5)

training accuracy: 0.7212

validation accuracy: 0.7103

testing accuracy: 0.71004

(training set: 0.3, validation set: 0.3, testing set: 0.4)

model = LogisticRegression(solver=' lbfgs')

training accuracy: 0.723

validation accuracy: 0.7086

testing accuracy: 0.7102

(training set: 0.3, validation set: 0.3, testing set: 0.4)

model = LogisticRegression(solver='newton-cg')

training accuracy: 0.7212

validation accuracy: 0.7113

testing accuracy: 0.7093

(training set: 0.3, validation set: 0.3, testing set: 0.4)

model = LogisticRegression(solver='liblinear')

training accuracy: 0.7212

validation accuracy: 0.7112

testing accuracy: 0.70905

(training set: 0.3, validation set: 0.3, testing set: 0.4)

model = LogisticRegression(solver='sag')

training accuracy: 0.7212

validation accuracy: 0.7112

testing accuracy: 0.7093

(training set: 0.3, validation set: 0.3, testing set: 0.4)

model = LogisticRegression(solver='saga')

training accuracy: 0.7213

validation accuracy: 0.7107

testing accuracy: 0.709

model = LogisticRegression(penalty='elasticnet', solver='saga', l1\_ratio=0.1)

training accuracy: 0.7210666666666666

validation accuracy: 0.7106

testing accuracy: 0.70935

model = LogisticRegression(penalty='elasticnet', solver='saga', l1\_ratio=0.3)

training accuracy: 0.721

validation accuracy: 0.7108666666666666

testing accuracy: 0.70975

model = LogisticRegression(penalty='elasticnet', solver='saga', l1\_ratio=0.5)

training accuracy: 0.7208666666666667

validation accuracy: 0.7106666666666667

testing accuracy: 0.7098

model = LogisticRegression(penalty='elasticnet', solver='saga', l1\_ratio=0.8)

training accuracy: 0.7212

validation accuracy: 0.7101333333333333

testing accuracy: 0.709

model = LogisticRegression(penalty='elasticnet', solver='saga', l1\_ratio=1.0)

training accuracy: 0.7210666666666666

validation accuracy: 0.7108

testing accuracy: 0.7088

stop\_words='english',

training accuracy: 0.7369333333333333

validation accuracy: 0.7283333333333334

testing accuracy: 0.72645

max\_features=None

training accuracy: 0.9343333333333333

validation accuracy: 0.8674666666666667

testing accuracy: 0.8696

ngram\_range=(2, 2)

training accuracy: 0.6671333333333334

validation accuracy: 0.6564

testing accuracy: 0.6544

ngram\_range=(1, 3)

training accuracy: 0.7018666666666666

validation accuracy: 0.6912666666666667

testing accuracy: 0.69235

binary=True

training accuracy: 0.6962666666666667

validation accuracy: 0.6873333333333334

testing accuracy: 0.68555

Q2

Max aggregation method

training accuracy: 0.8202292323112488

validation accuracy: 0.7882462739944458

testing accuracy: 0.7928499579429626

Q3

1. The aggregation method used by FastText.

Max aggregation method

training accuracy: 0.8202292323112488

validation accuracy: 0.7882462739944458

testing accuracy: 0.7928499579429626

Sum aggregation method

training accuracy: 0.5046419501304626

validation accuracy: 0.4978678226470947

testing accuracy: 0.4981499910354614

Mean aggregation method

training accuracy: 0.7543532252311707

validation accuracy: 0.7628153562545776

testing accuracy: 0.7638499736785889

1. The dimension of the embedding vectors

Dim = 32

training accuracy: 0.8165867328643799

validation accuracy: 0.6893435120582581

testing accuracy: 0.6880999803543091

Dim = 64

training accuracy: 0.8437944650650024

validation accuracy: 0.7887126803398132

testing accuracy: 0.7898499965667725

Dim = 128

training accuracy: 0.8733342289924622

validation accuracy: 0.8234274983406067

testing accuracy: 0.8240000009536743

Dim = 256

training accuracy: 0.8868381977081299

validation accuracy: 0.8363761305809021

testing accuracy: 0.8384000062942505

Dim = 512

training accuracy: 0.9040733575820923

validation accuracy: 0.836465060710907

testing accuracy: 0.8384999632835388

1. The optimizer, including learning rate, momentum or even algorithm

SGD lr=0.3

training accuracy: 0.8480810523033142

validation accuracy: 0.8246046900749207

testing accuracy: 0.8258499503135681

SGD lr=0.2

training accuracy: 0.8890813589096069

validation accuracy: 0.8397743701934814

testing accuracy: 0.8357499837875366

1. The number of epochs trained for

num\_epochs = 10

training accuracy: 0.9110029935836792

validation accuracy: 0.7701670527458191

testing accuracy: 0.7671999931335449

num\_epochs = 4

training accuracy: 0.8670042753219604

validation accuracy: 0.8169865012168884

testing accuracy: 0.8166999816894531

1. The batch size.
2. The preprocessor.