```
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.classification import LogisticRegressionWithSGD
from pyspark.mllib.feature import HashingTF, IDF

ham=sc.textFile("[path]/ham.txt")
spam=sc.textFile("[path]/spam.txt")



tf = HashingTF(numFeatures = 1000)
spamFeaturesRDD = spam.map(lambda email: tf.transform(email.split(" ")))
hamFeaturesRDD = ham.map(lambda email: tf.transform(email.split(" ")))

#create labeled point (label, featureVector) with corresponding classes: 1 for a spam, 0 for a safe
mails

positiveExamplesRDD = spamFeaturesRDD.map(lambda f: LabeledPoint(1, f))
negativeExamplesRDD = hamFeaturesRDD.map (lambda f: LabeledPoint(0, f))

#Put all data together

allTrainingData = positiveExamplesRDD.union(negativeExamplesRDD)

#Now train
spamClassModel = LogisticRegressionWithSGD.train(allTrainingData)

#Now predict on new data
posTestExample = tf.transform("O M G GET cheap stuff by sending money to ...".split(" "))
negTestExample = tf.transform("Hi Dad, I started studying Spark the other ...".split(" "))
anotherExample = tf.transform("O M G GET cheap stuff by sending money to  ..".split(" "))

print "Prediction for positive test example: %g" % spamClassModel.predict(posTestExample)
print "Prediction for negative test example: %g" % spamClassModel.predict(negTestExample)
print "Prediction for unknown test example: %g" % spamClassModel.predict(anotherExample)
```