Course <u>Progress</u> <u>Dates</u> **Discussion** Resources



Project due Oct 7, 2020 05:29 IST Completed

In this project you will be implementing linear classifiers beginning with the Perceptron algorithm. You will begin by writing your loss function, a hinge-loss function. For this function you are given the parameters of your model θ and θ_0 . Additionally, you are given a feature matrix in which the rows are feature vectors and the columns are individual features, and a vector of labels representing the actual sentiment of the corresponding feature vector.

Hinge Loss on One Data Sample

1.0/1 point (graded)

First, implement the basic hinge loss calculation on a single data-point. Instead of the entire feature matrix, you are given one row, representing the feature vector of a single data sample, and its label of +1 or -1 representing the ground truth sentiment of the data sample.

Reminder: You can implement this function locally first, and run python test.py in your sentiment_analysis directory to validate basic functionality before checking against the online grader here.

Available Functions: You have access to the NumPy python library as np; No need to import anything.

```
1 def hinge_loss_single(feature_vector, label, theta, theta_0):
      Finds the hinge loss on a single data point given specific classification
3
4
      parameters.
5
6
      Args:
7
           feature_vector - A numpy array describing the given data point.
8
          label - A real valued number, the correct classification of the data
9
              point.
10
          theta - A numpy array describing the linear classifier.
11
          theta 0 - A real valued number representing the offset parameter.
12
13
14
      Returns: A real number representing the hinge loss associated with the
15
      given data point and parameters.
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

```
CORRECT

See full output

See full output

You have used 3 of 25 attempts
```

The Complete Hinge Loss

1.0/1 point (graded)

Now it's time to implement the complete hinge loss for a full set of data. Your input will be a full feature matrix this time, and you will have a vector of corresponding labels. The k^{th} row of the feature matrix corresponds to the k^{th} element of the labels vector. This function should return the appropriate loss of the classifier on the given dataset.

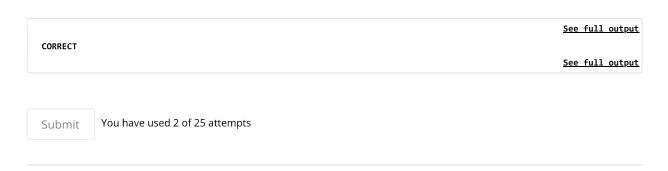
Available Functions: You have access to the NumPy python library as np, and your previous function as [hinge_loss_single]

```
Finds the total hinge loss on a set of data given specific classification
 4
      parameters.
 5
 6
      Args:
 7
          feature_matrix - A numpy matrix describing the given data. Each row
 8
               represents a single data point.
9
          labels - A numpy array where the kth element of the array is the
10
              correct classification of the kth row of the feature matrix.
11
          theta - A numpy array describing the linear classifier.
12
          theta_0 - A real valued number representing the offset parameter.
13
14
15
      Returns: A real number representing the hinge loss associated with the
       given dataset and nanameters. This number should be the average bing
```

Press ESC then TAB or click outside of the code editor to exit

Correct

Test results

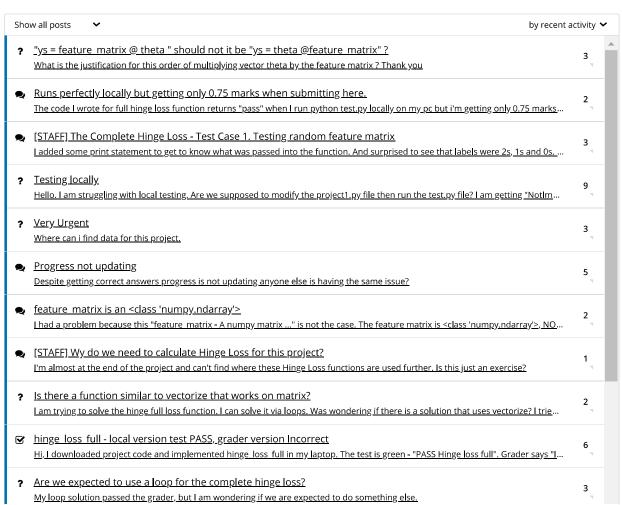


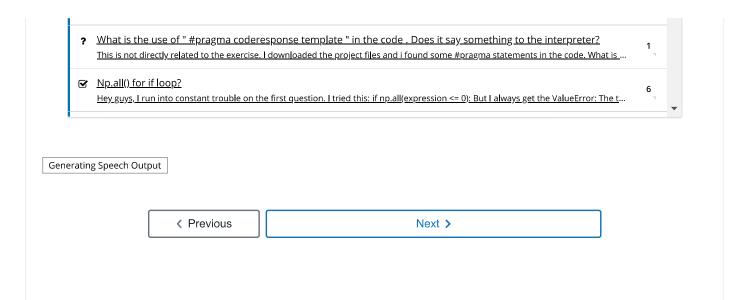
Discussion

Hide Discussion

Topic: Unit 1 Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 2. Hinge Loss

Add a Post





© All Rights Reserved



edX

About

Affiliates

edX for Business

Open edX

Careers

News

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

<u>Sitemap</u>

Connect

Blog

Contact Us

Help Center

Media Kit

Donate













© 2020 edX Inc. All rights reserved. 深圳市恒宇博科技有限公司 <u>粤ICP备17044299号-2</u>