

MITx 6.86x

Machine Learning with Python-From Linear Models to Deep Learning

Course **Progress** Discussion Dates Resources

A Course / Unit 1. Linear Classifiers and Generalizatio... / Project 1: Automatic R



 \square Bookmark this page

< Previous

Project due Mar 1, 2023 08:59 -03 Completed

Now we need a way to actually use our model to classify the data points. In this section way to classify the data points using your model parameters, and then measure the account of the section of the

Classification

1.0/1 point (graded)

Implement a classification function that uses θ and θ_0 to classify a set of data points. Ye matrix, θ , and θ_0 as defined in previous sections. This function should return a numpy a prediction is **greater than** zero, it should be considered a positive classification.

Available Functions: You have access to the NumPy python library as np.

Tip:: As in previous exercises, when $m{x}$ is a float, " $m{x}=m{0}$ " should be checked with $|m{x}|<$

```
1 def classify(feature_matrix, theta, theta_0):
2
      A classification function that uses theta and theta_0 to classify a se
3
      data points.
5
      Args:
6
          feature_matrix - A numpy matrix describing the given data. Each ro
7
              represents a single data point.
8
                   theta - A numpy array describing the linear classifier.
9
          theta - A numpy array describing the linear classifier.
10
          theta_0 - A real valued number representing the offset parameter.
11
      Returns: A numpy array of 1s and -1s where the kth element of the array
12
      the predicted classification of the kth row of the feature matrix using
13
      given theta and theta_0. If a prediction is GREATER THAN zero, it show
14
      be considered a positive classification.
15
```

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

Correct

Test results

CORRECT

Submit

THE PETASSELECT ACCOUNTS FINITE HOLD FOR THE CONTROL OF THE CONTRO

- a classifier function that, itself, takes arguments (feature_matrix, labels, **kwa
- the training feature matrix
- the validation feature matrix
- the training labels
- the valiation labels
- a **kwargs argument to be passed to the classifier function

This function should train the given classifier using the training data and then compute classification accuracy on both the train and validation data. The return values should I value is the training accuracy and the second value is the validation accuracy.

Implement classifier accuracy in the coding box below:

Available Functions: You have access to the NumPy python library as np, to classi already implemented and to accuracy which we defined above.

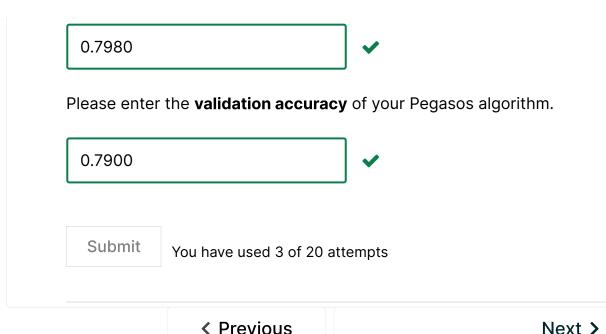
```
1 def classifier_accuracy(
 2
                   classifier,
 3
                   train_feature_matrix,
 4
                   val_feature_matrix,
 5
                   train_labels,
 6
                   val_labels,
 7
                   **kwargs):
           11 11 11
 8
 9
          Trains a linear classifier and computes accuracy. The classifier
           trained on the train data. The classifier's accuracy on the train
10
11
           validation data is then returned.
12
13
          Args:
                   `classifier` - A learning function that takes arguments
14
15
                            (feature matrix, labels, **kwargs) and returns (the
```

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

Correct

Test results

CORRECT



Discussion

Topic: Unit 1. Linear Classifiers and Generalizations (2 weeks):Project 1: Automatic Review Analyzer / 7. Classification and Accuracy



edX

<u>About</u>

Affiliates

edX for Business

Open edX

Careers

News

Legal

Terms of Service & Honor Code

Privacy Policy

Accessibility Policy

Trademark Policy

<u>Sitemap</u>

Cookie Policy

Do Not Sell My Personal Information

Connect

<u>Blog</u>