Midterm Exam

GITA315: Python Machine Learning

정보통신대학원

April 21, 2022

- starts at **7:10 pm** and ends at **7:50 pm**
- submit answers **before 7:55 pm**

Notice: Answer should be well structured and clear, easy to understand. Answers should be given in complete sentences. Any type of copy and paste should not be included in the answer.

Name: ____________________ Student Number: ______________________

100 points total

Problem 1: (30points total, 10 points each)

Consider a dataframe df with indexes (row numbers 0, 1, . . .) and labels (column names a, b, c, . . .). We also assume that the values of this dataframe are all integers.

a) explain df.iloc[: , [i, j]] for some valid integers, i and j.

Answer

b) explain df.iloc[[i, j], [p, q]] for some valid integers, i, j, p, and q.

Answer

c) describe the difference between df and df.values.

Answer

Problem 2: (30 points)

The following is the implementation of a perceptron in Python.

Describe, specifically, the operation of the block surrounded by the blue rectangle. Your answer *should* include the descriptions for the step-by-step operations in the block.

```
import numpy as np
class Perceptron(object):
    def __init__(self, eta=0.01, n_iter=50, random_state=1):
        self.eta = eta
        self.n_iter = n_iter
        self.random_state = random_state
    def fit(self, X, y):
        """
        Parameters
        ----------
        X : {array-like}, shape = [n_examples, n_features]
          Training vectors, where n_examples is the number of examples and
          n_features is the number of features.
        y : array-like, shape = [n_examples]
          Target values.
        ----------
        """
        rgen = np.random.RandomState(self.random_state)
        self.w_ = rgen.normal(loc=0.0, scale=0.01, size=1 + X.shape[1])
        self.errors_ = []
        for _ in range(self.n_iter):
            errors = 0
            for xi, target in zip(X, y):
                update = self.eta * (target - self.predict(xi))
                self.w_[1:] += update * xi
                self.w_[0] += update
                errors += int(update != 0.0)
            self.errors_.append(errors)
        return self
    def net_input(self, X):
        return np.dot(X, self.w_[1:]) + self.w_[0]
    def predict(self, X):
        return np.where(self.net_input(X) >= 0.0, 1, -1)
```

Answer:

Problem 3: (10 points)

We have two arrays, x and y:

n and *m* are integers, *n* > 1 and *m* > 1.

Shape of x is (*n*, *m*).

Shape of y is (*n*,).

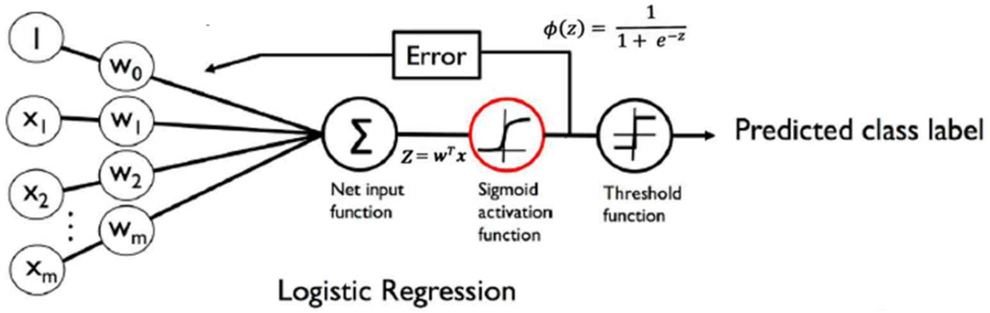
Any one of 1, 2, and 3 is selected for the value of y[*i*], i = 0, 1, 2, . . , *n*-1.

Explain x[y == 1, 1].

Answer:

Problem 4: (30 points)

The diagram below illustrates the concept of the logistic regression classifier. Explain the operation of this *in detail* using this diagram.



Answer: