

Multivariable Regression Questions

Multivariable Regression Overview: ➡ "please refer to notes"

How to find More Complex Boundaries:

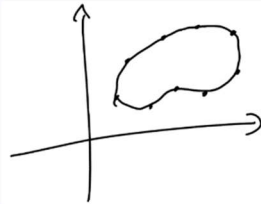
MCQ Questions:

1.

Predicting a Function

[Send Feedback](#)

A Prediction function like in the figure below to fit all training data will lead to



Options

This problem has only one correct answer

- ☒ Overfitting model
- ☐ Underfitting model
- ☐ Balanced model
- ☒ Hurray! Correct Answer

Solution Description

It will lead to overfitting, as it completely fits our training data, it remembers our training data points, and it will fail for the new dataset.

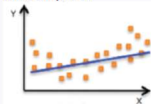
Attempts left: 1/2

2.

Figure is an Example Of

[Send Feedback](#)

Prediction function as shown in the figure below is an example of:



Options

This problem has only one correct answer

- ☐ Balanced
- ☒ UnderFit
- ☐ OverFit
- ☒ Hurray! Correct Answer

Solution Description

The prediction function as shown in the figure below is an example of underfitting. A better fit for this dataset is a quadratic equation.

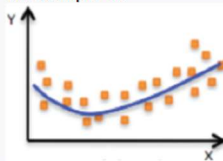
AI

3.

Figure is an Example Of

[Send Feedback](#)

Prediction function as shown in the figure below is an example of:



Options

This problem has only one correct answer

- ☒ Balanced
- ☐ UnderFit
- ☐ OverFit
- ☒ Hurray! Correct Answer

Solution Description

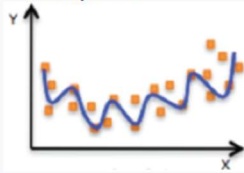
The prediction function as shown in the figure below is an example of balanced model.

4.

Figure is an Example Of

[Send Feedback](#)

Prediction function as shown in the figure below is an example of:



Options

This problem has only one correct answer

- ☐ UnderFit
- ☐ Balanced
- ☒ OverFit
- ☒ Hurray! Correct Answer

Solution Description

The prediction function as shown in the figure below is an example of overfitting.

Add Features to Boston Dataset:

In this problem you need to load the Boston dataset from sklearn toy datasets. After loading, you need to split the dataset into testing and training datasets. Now, fit the linear regression model on the training dataset and store the training and testing scores.

After this add two more columns to the dataset. These columns are the squared values of the 'AGE' and the 'RM' columns respectively.

Again split the dataset into testing and training datasets. Now, fit the linear regression model on the training dataset and store the training and testing scores.

Compare the scores of both the models and print "Score improved" if both training and testing scores improved after squaring two columns. Otherwise, print "Score not improved".

Output

If both training and testing scores improved after squaring two columns print:

"Score improved"

Else print:

"Score not improved"

Source Code:

```
3 import numpy as np
4 import pandas as pd
5 from sklearn.datasets import load_boston
6 from sklearn.model_selection import train_test_split
7 from sklearn.linear_model import LinearRegression
8 from sklearn.metrics import mean_squared_error, r2_score
9
10 # Load the Boston dataset
11 boston = load_boston()
12 data = pd.DataFrame(boston.data, columns=boston.feature_names)
13 target = pd.DataFrame(boston.target, columns=["MEDV"])
14
15 # Split the dataset into training and testing sets
16 X_train, X_test, y_train, y_test = train_test_split(data, target, test_size=0.2, random_state=42)
17
18 # Fit the linear regression model and store the training and testing scores
19 lr = LinearRegression()
20 lr.fit(X_train, y_train)
21 train_score_before = lr.score(X_train, y_train)
22 test_score_before = lr.score(X_test, y_test)
23
24 # Add squared columns for 'AGE' and 'RM'
25 X_train['AGE^2'] = X_train['AGE'] ** 2
26 X_train['RM^2'] = X_train['RM'] ** 2
27 X_test['AGE^2'] = X_test['AGE'] ** 2
28 X_test['RM^2'] = X_test['RM'] ** 2
29
30 # Fit the linear regression model with squared columns
31 lr.fit(X_train, y_train)
32 train_score_after = lr.score(X_train, y_train)
33 test_score_after = lr.score(X_test, y_test)
34
35 # Compare scores and print the result
36 if train_score_after > train_score_before and test_score_after > test_score_before:
37     print("Score improved")
38 else:
39     print("Score not improved")
40
```

Complexity Analysis of Normal Equation

Gradient Descent:

MCQ Questions:

1.

Gradient Descent Definition

[Send Feedback](#)

Gradient Descent is an optimization algorithm used for

Options

Attempts left: 1/2

This problem has only one correct answer

- ☒ minimizing the cost function in various machine learning algorithms
- ☐ Certain Changes in algorithm
- ☐ maximizing the cost function in various machine learning algorithms
- ☐ remaining same the cost function in various machine learning algorithms
- ☒ Hurray! Correct Answer

Solution Description

Gradient Descent is an optimization algorithm used for minimizing the cost function in various machine learning algorithms

2.

Cost in Gradient Descent

[Send Feedback](#)

Following is the equation to calculate the cost in Linear Regression. In Gradient Descent our task is to:

$$y^p = mx + c$$

$$cost = \sum_i (y_i - (mx_i + c))^2$$

Options

This problem has only one correct answer

- ☐ Find m and c to maximize the cost
- ☒ Find m and c to minimize the cost
- ☐ Make the cost zero
- ☒ Hurray! Correct Answer

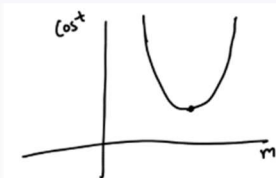
Solution Description

In Gradient Descent our task is to find m and c to minimize the cost.

3.

Cost Function

[Send Feedback](#)



In above graph between cost and m, dot on parabola represent:

Options

This problem has only one correct answer

- ☐ It is a point where cost is zero
- ☒ It is a point where cost is minimum
- ☐ It is a point where cost is maximum
- ☒ Hurray! Correct Answer

Solution Description

It is the lowest point in the parabola so the cost is minimum.

4.

Cost Function

[Send Feedback](#)

Choose all correct statements:



Options

This problem may have one or more correct answers

- ☐ Left tangent on parabola is having +ve slope
- ☐ Right tangent on parabola is having -ve slope
- ☒ Left tangent on parabola is having -ve slope✓
- ☒ Right tangent on parabola is having +ve slope✓
- ☒ Hurray! Correct Answer

Solution Description

1. Left tangent on the parabola is having the -ve slope.
2. Right tangent on the parabola is having +ve slope.

Learning Rate:

MCQ Questions:

1.

Gradient Descent

[Send Feedback](#)

You run gradient descent for 20 iterations with $\alpha=0.3$ and compute cost function after each iteration. You find that the value of cost function decreases slowly and is still decreasing after 20 iterations. Based on this, which of the following conclusions seems most plausible ?

Options

Atte

This problem has only one correct answer

- ☐ It'd be more promising to try a smaller value of α (say $\alpha=0.1$)
- ☒ It'd be more wise to try a larger value of α (say $\alpha=1.0$)
- ☐ $\alpha=0.3$ is an effective choice of learning rate.

The solution to this problem has been viewed

Solution Description

A larger value for α should increase the rate of convergence to the minimum of $J(\theta)$.

2.

Learning Rate

[Send Feedback](#)

If learning rate is high:

Options

A

This problem has only one correct answer

- ☒ There is a risk Overshooting the lowest point.
- ☐ It is always better to have high learning rate
- ☐ It will slow down the minimization process
- ☒ Hurray! Correct Answer

Solution Description

If the learning rate is high then there is a risk Overshooting the lowest point.

3.

Learning Rate

[Send Feedback](#)

Choose the correct option for the statement below:

With a very low learning rate:

Options

Attempts left: 1/2

This problem has only one correct answer

- ☐ A low learning rate is more precise.
- ☐ Calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.
- ☒ All
- ☒ Hurray! Correct Answer

Solution Description

With a very low learning rate:

1. A low learning rate is more precise.
2. Calculating the gradient is time-consuming, so it will take us a very long time to get to the bottom.

Code Gradient Descent: → "please refer to notes"

Code Gradient Descent:

MCQ Questions:

1.

N Feature Gradient Descent

[Send Feedback](#)

As mentioned in the video what all are the changes we need to make in above function to make it applicable for n features?

```
def step_gradient(points, learning_rate, m, c):  
    m_slope = 0  
    c_slope = 0  
    N = len(points)  
    for i in range(N):  
        x = points[i][0]  
        y = points[i][1]  
        m_slope += (-2/N) * (y - m * x - c) * x  
        c_slope += (-2/N) * (y - m * x - c)  
    new_m = m - learning_rate * m_slope  
    new_c = c - learning_rate * c_slope  
    return new_m, new_c
```

Options

Attempts left: 0/2

This problem may have one or more correct answers

- ☒ The way we are calculating m_slope and c_slope in the function above need not to be changed.
- ☐ In function signature, instead of m and c, an array is passed having all m and last col as c ✓
- ☒ Function signature need not be changed.
- ☒ Instead of m_slope & c_slope we will have an array that starts with all 0 ✓

The solution to this problem has been viewed

Solution Description

1. In function signature, instead of m and c, an array is passed having all m and last col as c.
2. Instead of m_slope & c_slope we will have an array that starts with all 0.

Variations of Gradient Descent

MCQ Questions:

1.

Gradient Descent Types:

[Send Feedback](#)

Select all variations of gradient descent that are discussed in the video ?

Options

This problem may have one or more correct answers

- ☒ Batch ✓
- ☐ Silly
- ☐ Lilly
- ☒ Mini ✓
- ☒ Stochastic ✓
- ☒ Hurray! Correct Answer

Solution Description

1. Batch
2. Mini
3. Stochastic

2.

Which Gradient Descent

[Send Feedback](#)

Fill in the blanks ?

We compute the cost gradient based on the complete training set; hence, we sometimes also call it _____.
In case of very large datasets, using this gradient descent can be quite costly since we are only taking a single step for one pass over the training set.

Options

Attempts left: **1/2**

This problem has only one correct answer

- ☒ Batch gradient descent
- ☐ Stochastic gradient descent
- ☐ Mini gradient descent
- ☒ Hurray! Correct Answer

Solution Description

We compute the cost gradient based on the complete training set; hence, we sometimes also call it Batch gradient descent. In the case of very large datasets, using this gradient descent can be quite costly since we are only taking a single step for one pass over the training set.

3.

Match Statement to Types of Gradient Descent

[Send Feedback](#)

Do proper match for below statements:

1. Uses n data points instead of 1 sample at each iteration.
2. Computes the gradient using a single sample.
3. Computes the gradient using the whole dataset.

Types of Gradient Descent

- A: Mini-batch gradient descent
B: Stochastic gradient descent
C: Batch gradient descent

Options

This problem may have one or more correct answers

- ☒ 3 --- > C ✓
- ☐ 3 --- > B
- ☒ 2 --- > B ✓
- ☐ 2 --- > A
- ☒ 1 --- > A ✓
- ☐ 3 --- > A
- ☐ 1 --> C

☒ Hurray! Correct Answer

Solution Description

Mini-batch gradient descent: Uses n data points instead of 1 sample at each iteration.

Stochastic gradient descent: Computes the gradient using a single sample.

Batch gradient descent: Computes the gradient using the whole dataset.

→ Gradient Descent Notes

→ Multivariable Regression And Gradient Descent Notebook