

zID: z5224151

Name: Zanning Wang

Q1.

HATEOAS is an improved application layer architecture based on traditional REST with certain restrictions. With this architecture, the server can dynamically feed back client information, and the front and back ends can be updated iteratively. At the same time, users do not need to know how to interact with the application layer in depth. One of best example is that the spring support HATEOAS to help develop the frontend coding.

Q2.

In most cases, deep learning is suitable for large-scale prediction projects with a large amount of data. It can better abstract the characteristics of the learning object, compared with traditional machine learning.

And deep learning needs to dynamically adjust each neuron. For models that are low-cost or pursue training time, the deep learning is not fittable.

Q3.

- Use with SSL
- Token should be expired after certain time
- If authorized, no extra lookup required

Example: during the user login process, we should build a SSL connect to give user token to store user session.

Q4.

For John and Smith, the student ID are not unique, and they have different name format.

For Ada wong, his Age column is negative (wrong data value)

For the Date of birth column, the data type are not consistent

For the City columns, there is a misspelling for Sydney (wrong data value)

Q5.

Q6.

(A)

In this case, I think k-means is more suitable. k-means can help us divide users into different clusters. We can classify subsequent users into different clusters based on similarity to infer

user preferences

(B)

We first need to randomly initialize k cluster centers. And divide each point to the center closest to them and classify it into one cluster. Then recalculate the center of a cluster, according to the newly drawn point. Repeat the above steps to know that no new center point will be generated

(C)

Set the starting value k to 3, and then gradually increase k to see which situation is best for classification.

Q7.

Precision = $TP/(TP+FP) = 8/(8+2) = 0.8$

Recall = $TP/(TP+FN) = 8/(8+12) = 0.4$

F1-score = $2 * \text{precision} * \text{Recall} / (\text{Precision} + \text{Recall}) = 2 * 0.8 * 0.4 / (0.8 + 0.4) = 0.53$

Therefore the precision is 0.8, the recall is 0.4 and the F1-score is 0.53

Q8.

A:

HTTP/1.1 201 Created

DATE: (detailed time)

Content Type: text/html

Content length:

Content: New URI

Expires: (detailed time)

B:

HTTP/1.1 200 OK

DATE: (detailed time)

Content Type: text/html

Content length:

Content: [](a list of order which have sorted according to the requirement)

Expires: (detailed time)

Q9.

(1)

According to the current situation, we should first reconfirm the repair order of the problem according to the severity and ensure that the older version is repaired as soon as possible.

(2)

I may use pandas to preprocessing the data, we should filter some null value, and ensure each row's data is clean

(3)

I may choose the KNN models to solve this problem, by using the KNN, we may classificate

different vulnerability into one cluster and first solve the cluster which Vulnerability Severity Score is the highest.

Q10.

(1)

I think this recommendation system is more likely to be user-based collaborative filtering. From the database given, There are some movies that have very few reviews. It is difficult to find users who are close to them if you recommend them based on items with less evaluation. It may also cause cold start problems.

(2)

The system will give priority to finding people with similar interests, that is, people who are closer to it. Then go to see which movies similar users have watched, and recommend the ones that the user did not watch to him. If the number of user reviews is small, it will be difficult to find users who are similar to him, resulting in inaccurate recommendations. You can use the previous recommendation of some popular movies to get the user' s initial evaluation.