

🔗 Google Discussions



Exam Professional Data Engineer All Questions

View all questions & answers for the Professional Data Engineer exam

[Go to Exam](#)

📄 EXAM PROFESSIONAL DATA ENGINEER TOPIC 1 QUESTION 89 DISCUSSION

Actual exam question from Google's Professional Data Engineer

Question #: 89

Topic #: 1

[\[All Professional Data Engineer Questions\]](#)

You're training a model to predict housing prices based on an available dataset with real estate properties. Your plan is to train a fully connected neural net, and you've discovered that the dataset contains latitude and longitude of the property. Real estate professionals have told you that the location of the property is highly influential on price, so you'd like to engineer a feature that incorporates this physical dependency.

What should you do?

- A. Provide latitude and longitude as input vectors to your neural net.
- B. Create a numeric column from a feature cross of latitude and longitude.
- C. Create a feature cross of latitude and longitude, bucketize it at the minute level and use L1 regularization during optimization.
- D. Create a feature cross of latitude and longitude, bucketize it at the minute level and use L2 regularization during optimization.

[Show Suggested Answer](#)

by [nwk](#) at *Sept. 2, 2022, 10:26 a.m.*

Comments



Type your comment...

[Submit](#)

  **AHUI** Highly Voted  2 years, 7 months ago

Ans C, use L1 regularization because we know the feature is a strong feature. L2 will evenly distribute weights

   upvoted 9 times

  **dish11dish** Highly Voted  2 years, 5 months ago

Selected Answer: C

Option C is correct

Use L1 regularization when you need to assign greater importance to more influential features. It shrinks less important feature to 0.
L2 regularization performs better when all input features influence the output & all with the weights are of equal size.

   upvoted 8 times

  **desertlotus1211** Most Recent  1 month, 3 weeks ago

Selected Answer: D

L1 regularization (Option C) would encourage sparsity but may eliminate too many features, which can be detrimental when you need to capture subtle geographic differences



   upvoted 2 times

  **SamuelTsch** 6 months, 2 weeks ago

Selected Answer: D

I would like choose D. L1 will ignore the irrelevant features. However, we know that lat and long are crucial for this model. We can't take away their influences. L2 helps in preventing overfitting.


   upvoted 3 times

  **MohaSa1** 6 months, 3 weeks ago

Selected Answer: A

This does not seem to be useful, minute level bucketizing will create 3,600 possible buckets per degree squared, not logical, and sparse feature space, Option A seems to be a better choice.

   upvoted 1 times

  **Snnnnneee** 9 months, 2 weeks ago

Selected Answer: B

Bucketing into minutes is inaccurate, up to 1.8 km are grouped. Way too much for real estate.
Therefore B

   upvoted 1 times

  **uday_examtopic** 1 year, 7 months ago

Create a feature cross of latitude and longitude, bucketize it at the minute level and use L2 regularization during optimization.

Like option C, we bucketize at the minute level, but this time we apply L2 regularization. L2 regularization, or Ridge Regression, discourages large values of weights in the model without forcing them to become sparse. It can help prevent overfitting, especially when we have a large number of features (as a result of bucketizing and crossing).

Given the options, D. Create a feature cross of latitude and longitude, bucketize it at the minute level and use L2 regularization during optimization seems to be the most appropriate. Bucketizing at the minute level captures localized patterns, and L2 regularization can help control the complexity of the model without enforcing sparsity.

   upvoted 2 times

  **ckanaar** 1 year, 7 months ago

What does bucketizing at the minute level mean in the context of this question?

   upvoted 3 times

  **Surely1987** 1 year, 5 months ago




Coordinates are written with Degrees, minutes and seconds (one minute being equal to about 1.8 km). So you group your coordinates in buckets with a minute precision

   upvoted 4 times

  **FP77** 1 year, 8 months ago

Selected Answer: B

I strongly believe it's B.

   upvoted 2 times

  **Mathew106** 1 year, 9 months ago

Selected Answer: B

The right answer is B. What the hell does bucketize the feature cross of latitude and longitude even mean? They are not a time feature. C and D don't even make sense. The L1 regularization is something that doesn't answer anything in the question. The only valid feature engineered here is option B. A is not an engineered feature.

Create a feature cross of latitude and longitude, bucketize it at the minute level and use L1 regularization during optimization.

👍 ↩ 🚩 upvoted 1 times

🗨️ 👤 **baimus** 7 months, 2 weeks ago

Bucketising means that we're saying "anyone in this square 1.8km (minute) region is considered a single area" - it's actually recommended as a default way to deal with lat/lon, as they don't really work as separate columns (or at least we'd be hoping the FCNN buckets them intelligently itself, which it won't mostly)

👍 ↩ 🚩 upvoted 1 times

🗨️ 👤 **Jojo9400** 1 year, 9 months ago

D

You have to use L2, since you have create a new variable with two already existing the risk of multicollinearity is high, L1 is good for selecting feature to avoid curse of dimensionality not for multicollinearity

👍 ↩ 🚩 upvoted 1 times

🗨️ 👤 **ga8our** 2 years ago

Why not L2? L2 (Ridge) uses a squared value coefficient as a penalty term to the loss function, while L1 (Lasso) uses an absolute value coefficient. Isn't a squared penalty stronger than an absolute one?

<https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c>

👍 ↩ 🚩 upvoted 1 times

🗨️ 👤 **ckanaar** 1 year, 7 months ago

L1 regression forces unimportant coefficients to zero. Since the location is extremely important, L1 will force less important coefficients to zero, thereby further increasing the importance of the location coefficient.

👍 ↩ 🚩 upvoted 2 times

🗨️ 👤 **Oleksandr0501** 2 years ago

gpt: Option C and D suggest bucketizing the feature cross of latitude and longitude at the minute level and using L1 or L2 regularization during optimization. While regularization can help prevent overfitting, bucketizing at such a granular level may not be necessary and could lead to overfitting. It's also not clear how bucketizing at the minute level would capture the spatial relationship between the latitude and longitude features.

👍 ↩ 🚩 upvoted 2 times

🗨️ 👤 **PolyMoe** 2 years, 3 months ago

Selected Answer: D

D. Create a feature cross of latitude and longitude, bucketize it at the minute level and use L2 regularization during optimization. This will create a new feature that captures the physical dependency of the location of the property on the price, and bucketing it at the minute level will reduce the number of unique values and prevent overfitting. L2 regularization will also help to prevent overfitting by penalizing large weights in the model.

👍 ↩ 🚩 upvoted 2 times

🗨️ 👤 **cetanx** 1 year, 11 months ago

chat-gpt also says D

explanation:

This approach effectively creates a grid of the geographical area in your data, allowing the model to learn weights for each grid cell (bucket). This helps capture the spatial relationship between latitude and longitude, which can be crucial for real estate prices. Additionally, using L2 regularization helps prevent overfitting by discouraging complex models, which can be particularly important when working with high-dimensional crossed features.

👍 ↩ 🚩 upvoted 1 times

🗨️ 👤 **zelck** 2 years, 5 months ago

Selected Answer: C

C is the answer.

<https://developers.google.com/machine-learning/crash-course/feature-crosses/video-lecture>

A feature cross is a synthetic feature formed by multiplying (crossing) two or more features. Crossing combinations of features can provide predictive abilities beyond what those features can provide individually.






<https://developers.google.com/machine-learning/crash-course/regularization-for-sparsity/l1-regularization>






👍 ↩ 🚩 upvoted 3 times






🗨️ 👤 **crismo04** 2 years, 7 months ago

<https://medium.com/riga-data-science-club/geographic-coordinate-encoding-with-tensorflow-feature-columns-e750ae338b7c#:~:text=to%20the%20rescue!-,Feature%20Crosses,-Combining%20features%20into>

👍 ↩ 🚩 upvoted 2 times

  **crismo04** 2 years, 7 months ago
Feature cross seems to be the right feature option
   upvoted 1 times

  **crismo04** 2 years, 7 months ago
So it's B option
   upvoted 4 times

  **[Removed]** 2 years, 8 months ago
Selected Answer: C
Regularization + location into one
   upvoted 1 times
[Load full discussion...](#)



Platform

> [Home](#)

> [All Exams](#)

> [Examtopics PRO](#)

> [Training Courses](#)

