

- . In the mapping of categorical values, can models directly multiply strings by the learned weights?

1 / 1 punto

☒ No☐ Yes☒ **Correcto**

Correct! Feature engineering must be applied in advance to convert strings into numerical values.

2. You are working on a taxi tip prediction model and your raw dataset has columns for the latitude and longitude of both pickup and dropoff locations. These do not assume a Gaussian distribution. Among the choices below, which two approaches are more likely to make your model learn better from these features? Assume that you are just starting the feature engineering process. (Select two answers)

1 / 1 punto

☐ Because the data does not assume a Gaussian distribution, you should normalize these location features following the formula:

$$X_{norm} = (X - X_{min}) / (X_{max} - X_{min})$$

This puts the values into the range [0,1] so it can help the training converge faster.

☒ Bucketize the locations into discrete bins then do a feature cross of the latitude and longitude.

☒ **Correcto**

Correct! Bucketizing then crossing the features can possibly enhance the predictive quality of the data. For example, a *latitude x longitude* cross of the pickup locations can let the model learn which combination of these features yield to a bigger tip.

☐ Do not consider these features because you noticed that the raw data has a column for trip distance in kilometers. Your intuition says this has greater impact on the tip amount compared to the locations.

☒ Bucketize the locations into discrete bins.

☒ **Correcto**

Correct! Bucketizing the coordinates will group close proximities together. With this technique, the model can learn which areas are likely to give a bigger tip.

3. Which of the following should you implement when serving your model to ensure its performance?

1 / 1 punto

- ☐ Increase the predictive quality of your data through feature engineering techniques such as scaling, normalization, standardization, bucketizing, etc.
- ☐ Ensure that each request is processed in real-time.
- ☒ Make sure all data transformations are the same in any scenario
- ☐ Use as few features as possible to save compute resources



Correcto

Good job! The same transformations you did during feature engineering should be applied to the user input when serving your model. This is to avoid training-serving skews. For example, if a feature is normalized during training, then the serving input should also be normalized. Else, the model might behave unexpectedly.