Module Quiz

Quiz, 6 questions

1 point

1. True or False - In ML, you could train using all your data and decide not to hold out a test set and still get a good model

- (●) True
- ( ) False

---

1 point

2. You are tasked with splitting your dataset into 80% training and 20% evaluation for your ML model. Your partner wrote the below SQL script for you to use. Should you use it to create your datasets? Why or why not

```
#standardSQL
WITH
  alldata AS (
  SELECT
    IF (RAND() < 0.8,
      'train',
      'eval') AS dataset,
    arrival_delay,
    departure_delay
  FROM
    `bigquery-samples.airline_ontime_data.flights`
  WHERE
    departure_airport = 'DEN'
    AND arrival_airport = 'LAX' ),
  training AS (
```

Module Quiz

Quiz, 6 questions

1 point

2. You are tasked with splitting your dataset into 80% training and 20% evaluation for your ML model. Your partner wrote the below SQL script for you to use. Should you use it to create your datasets? Why or why not

```
#standardSQL
WITH
  alldata AS (
  SELECT
    IF (RAND() < 0.8,
      'train',
      'eval') AS dataset,
    arrival_delay,
    departure_delay
  FROM
    `bigquery-samples.airline_ontime_data.flights`
  WHERE
    departure_airport = 'DEN'
    AND arrival_airport = 'LAX' ),
  training AS (
  SELECT
    SAFE_DIVIDE( SUM(arrival_delay * departure_delay) , SUM(departure_delay *
      departure_delay)) AS alpha
  FROM
    alldata
  WHERE
    dataset = 'train' )
```

- Yes use it - the RAND() function is only called once at the very beginning so all the data is put into training and evaluation in a repeatable fashion
- Almost - instead of using a WITH clause, if you stored the training and testing data permanently then you will always have the exact same dataset to train and evaluate on.
- Yes use it - this will yield 80% training, 20% validation due to the RAND() filter logic
- [x] No - the use of RAND(), even if only called once to divide the training and validation dataset, makes the experiment not repeatable for anyone else trying to start with the same datapoints. Consider using a hash function and a modulo operator instead.

---

1 point

3. What is a way to approximate or model real world unknown data? (choose all that apply)

- [ ] You can't because real world data cannot be modeled
- [x] Increase the breadth and quality of the data you have available. The better the data, the easier it will be for the model to learn.
- [ ] Split your dataset into three training buckets: training, validation, and production. Train the model on each of the three datasets to find which dataset

- [x] Increase the breadth and quality of the data you have available. The better the data, the easier it will be for the model to learn.
- [ ] Split your dataset into three training buckets: training, validation, and production. Train the model on each of the three datasets to find which dataset performs the best. Then, use only that dataset to train your production model.
- [x] Split your dataset into separate buckets and train your model only on a portion of that dataset (keeping the rest as held out which will model unseen data)

---

1 point

4. What's a recommended way to split your dataset in a repeatable fashion using SQL?

- ( ) Use a hash function and a RAND() function
- ( ) Use a RAND() function to ensure a random sample
- (•) Use a modulo operator and a hash function
- ( ) There is no way to get the same data values again
- ( ) Use a modulo operator and a RAND() function

Module Quiz

Quiz, 6 questions

1 point

5. Check all the common pitfalls for splitting a dataset even if done properly:

- [ ] Your validation dataset will never be as good as your training dataset
- [x] You can no longer predict using the field you split the data on
- [x] Your splitting field may not be noisy enough for granular divides of your dataset
- [x] You might not have enough data to split the dataset into training, validation, and testing

1 point

6. What can you do if your model passes validation but fails testing?

(Select all 2 correct answers)

- [x] Start from the beginning with a brand new model type
- [x] Stop model training and work to collect new data points before trying the same model again
- [ ] Re-train the same model again with different hyperparameters to optimize for a lower loss metric on your testing dataset