**1.**

Question 1

What are the three key components we should consider when serving an ML Model in a production environment? (Select all that apply)

**1 / 1 point**



Input Data

**Correct**

You’ve got it! The model executed on-device makes predictions based on the input data.



An interpreter

**Correct**

Right on track! An Interpreter encapsulates a pre-trained model in which operations are executed for inference.



An orchestrator



A model

**Correct**

Correct! Providing the algorithm and training the ML model is the first step towards putting it into production.

**2.**

Question 2

What happens after a while in operation to an offline-trained model dealing with new real-live data?

**1 / 1 point**



The model adapts to new patterns.



The model abruptly forgets all previously learned information.



The model becomes stale.

**Correct**

Good job! The model performance deteriorates to the point of the model not being any longer fit for purpose. This phenomenon is called model decay and should be carefully monitored.

**3.**

Question 3

In applications that are not user-facing, is throughput more critical than latency for customer satisfaction?

**1 / 1 point**



Yes, in this case, we are concerned with maximizing throughput with the lowest CPU usage.



No, because users might complain that the app is too slow.

**Correct**

Correct! Latency is not a key concern for back-end services.

**4.**

Question 4

Nowadays, developers aim to minimize latency and maximize throughput in customer-facing applications. However, in doing so, infrastructure scales and costs increase. So, what strategies can developers implement to balance cost and customer satisfaction? (Select all that apply)

**1 / 1 point**



GPU sharing

**Correct**

Nailed it! This strategy reduces the cost of GPU-accelerated computing.



Multi-model serving

**Correct**

Yes! This approach scales back infrastructure.



Optimizing inference models

**Correct**

Right on track! Optimization modifies a model to handle a higher load, reducing costs as a result.



Stress testing