1. Describe the publish/subscribe model as it pertains to events, messaging systems, producers, consumers, and message brokers. (2 pt)

A producer (the starting point of a message) creates an event, which is then asynchronously sent to a message broker (once the event data is sent, the producer forgets about it). The producer is also known as a publisher. The message broker then sends the event body to all subscribers, which are the end consumers of the message contained in the event. The name *subscriber* is derived from the fact that each consumer is subscribed to a topic, or particular stream of messages. If consumers are unreachable or messages are being transmitted faster than consumers can process them, the message broker may use backpressure to stop the producer from sending new messages or write messages to disk until the queue is more manageable.

1. What are the two main patterns of messaging when using multiple consumers? (2 pt)

The two main patterns of messaging when there are multiple consumers are either through a message broker, or brokerless (direct from publisher to consumer). The former system is described in the prior question, whereas the latter system uses a network to distribute messages directly from producers to consumers. With brokerless messaging, however, the producer must send the event separately to each consumer.

1. Explain the concepts of change data capture and event sourcing, and how they differ. (2 pt)

Change data capture (CDC) logs the changes of a database to a log in an append-only fashion asynchronously. The logs are applied to the database index(es) so that once the changes are applied, the partitioned indexes are up to date with the current state of the database. CDC works with the database as mutable items, so the application working with the database is not involved in the changes made.

Event sourcing is similar to CDC, except that this reflects all changes applied to a database, as opposed to some outright updates/deletes that CDC implements. This makes retrospective lookbacks at a database possible. Since this log is immutable, event sourcing makes changes on an application level.

1. What are Complex Event Processing and Stream Analytics, and how do they differ? (2 pt)

Complex Event Processing (CEP) is an approach that uses a declarative query language (like SQL, according to Kleppmann) to identify matching events that a consumer will then process. This is useful for selective event processing by consumers, avoiding or supplementing topic subscriptions.

Stream analytics differ from CEP in that they are what they sound like – combining aggregates over some fixed window. Stream analytics can use a tumbling window, which has a fixed length and which separates events. An event cannot belong to more than one tumbling window. A sliding window is a continuous window that changes, but keeps all events that happen within a certain time interval. Hopping windows are a middle ground between sliding and tumbling windows, where events are retained within a fixed length, but the windows overlap.

1. What are some machine learning algorithms that use greedy algorithms? (2 pt)

Machine learning algorithms that employ greedy algorithms include tree-based learning (CART, Random Forests) and nearest neighbor algorithms. Tree-structured learning uses greedy algorithms to trim the tree, while nearest neighbor algorithms find a local optimization by finding the closest approximation for a data point’s class. Realistically, any local optimization that occurs in machine learning should relate back to a greedy algorithm at its base.