Q1. How does the Gradient-Boosted Tree Algorithm work in Classification? How does Gradient Boost differ from AdaBoost and Logistic Regression?

A1. Boosting works on the principle of improving mistakes of the previous learner through the next learner.. Boosting focuses on sequentially adding up these weak learners and filtering out the observations that a learner gets correct at every step.

Gradient boosting works by building simpler (weak) prediction models sequentially where each model tries to predict the error left over by the previous model. In the Classification model, the target variable is binary, here the loss function changes to log-likelihood, so in the first step to initialize the model with some constant value, we use log(odds). Then we find residuals, after this we build a DT. we find the final output of the leaves because there might be a case where a leaf gets more than 1 residuals, so we need to calculate the final output value

Ada Boost - An additive model where shortcomings of previous models are identified by high-weight data points. The trees are usually grown as decision stumps. Each classifier has different weights assigned to the final prediction based on its performance.

Logistic Regression - The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

Q2. What is a Delta Lake and how does it offer a solution to building reliable data pipelines?

A2. Delta Lake is the optimized storage layer that provides the foundation for storing data and tables in the Databricks Lakehouse Platform. It provides following features to build reliable data pipelines - (credits analytics vidhya blog post)

- Users can access the metadata using the Describe Detail feature, and it is stored in the same way as other data.
- It examines each column, data type, etc. in the Schema, which is read as part of the metadata.
- A single architecture is offered by Delta Lakes for reading both batch and stream data.

Q3. When working with Pandas, we use the class pandas.core.frame.DataFrame and when working with the pandas API in Spark, we use the class pyspark.pandas.frame.DataFrame, are these the same, explain why or why not?

A3. No, they are not same, pandas.core.frame.Dataframe is a class under Pandas library which allows data wrangling or compute on the datasets on a single machines, while pyspark.pandas.frame.Dataframe is a class from pandas API in spark, which allows datasets wrangling or compute in distributed computing cluster (Hadoop). Hence difference arise in single node vs multiple node as well as the execution methodology of both of these (Eager Execution Vs Lazy Execution)

Pandas DataFrame is a potentially heterogeneous two-dimensional size-mutable tabular data structure with labeled axes (rows and columns). In Spark, DataFrames are distributed data collections that are organized into rows and columns. Each column in a DataFrame is given a name and a type.

Q4. What is a Machine Learning Pipeline and why is it important? What are the steps in a Machine Learning workflow?

A4. The splitting machine learning task into multi-steps in independent, reusable and modular parts , so that they acn be pipelined together to create models, hence to make it easy to scalable as well as use in different uses as or when need arises for them. These tasks include preprocessing, feature scaling and feature selections. The steps are connected and executed in well defined order. This type of ML pipeline makes building models more efficient and simplified, cutting out redundant work

This enables to automate the ML workflow by enabling data transformation, correlation and this can be utilized to achieve output. With the ML pipeline, each part of your workflow is abstracted into an independent service. Then, each time you design a new workflow, you can pick and choose which elements you need and use them where you need them, while any changes made to that service will be made on a higher level.