**What is Normalization and how does Normalization make training a model more stable?**

Normalization scales dataset features to a mean of zero and a standard deviation of one. Controlling the magnitude of input features and avoiding one feature from dominating others stabilizes machine learning model training. Features with varying scales can have a greater impact on the model than others, causing instability and possibly inferior outcomes. Normalizing the data ensures that all characteristics have the same scale, which improves model convergence and stability during training.

**What are loss and optimizer functions and how do they work?  
  
What is Gradient Descent and how does it work?**

A machine learning model's loss and optimizer functions are crucial.

A loss function, also called a cost function, measures the difference between the model's expected and actual outputs (target labels). The model minimizes loss function. Cross-entropy, hinge loss, and mean squared error are loss functions.

The model's parameters are modified by the loss function using an optimizer function. Gradient descent adjusts parameters to minimize loss function. SGD, Adam, and Adagrad optimizer algorithms compute gradients and update model parameters differently.

Gradient Descent optimizes model parameters to minimize a cost or loss function in machine learning. Iteratively updating model parameters in the steepest loss function drop direction works.

Each iteration, the gradient of the loss function with respect to the model parameters is calculated and used to update the parameters to minimize the loss function. A learning rate hyperparameter controls the update size at each iteration.

**What is an activation function?  
What are the outputs of the following activation functions: ReLU, Softmax Tanh, Sigmoid**

An activation function is a mathematical function that is applied to the output of each node (neuron) in a neural network. The activation function's aim is to add non-linearity into the node's output, allowing the neural network to learn a broader variety of functions and reflect complex data distributions.

* ReLU (Rectified Linear Unit): The ReLU activation function outputs the input value if positive and 0 otherwise. Mathematically, y = max(0, x) is the activation function's input and output.
* Softmax: The final neural network layer for multi-class classification uses the Softmax activation function. It generates a probability distribution over the classes that sums to 1. Mathematically, it is y i = e^(x i) / \_j(e^(x j)), where x is the activation function input vector and y is the output vector.
* Tanh (Hyperbolic Tangent): The Tanh activation function outputs [-1, 1]. It adds non-linearity to neural network hidden layers. Mathematically, y = tanh(x) is the activation function's output.
* Sigmoid: For binary classification tasks, the Sigmoid activation function outputs values between 0 and 1. It converts input into a probability-like value, with values near to 1 suggesting a high possibility of a positive class and values close to 0 indicating a negative class. Mathematically, y = 1 / (1 + e^(-x)) where x is the activation function input and y is the output.

**What is the TPOT algorithm and how does it work?    
What does TPOT stand for?**

TPOT ((Tree-based Pipeline Optimization Tool) functions by creating a number of candidate pipelines and evaluating their performance on training data using cross-validation. Pipelines are depicted as trees, with internal nodes representing activities (such as feature engineering or feature selection) and leaves representing machine learning algorithms.