NAME-RAFIUR RAHMAN MASHRAFI

STUDENT ID: 2221971042

CSE445.4

Part A: Fundamentals of Learning

1.Supervised Learning: when a computer algorithm trained on data that already labeled or categorized. After training phase it can apply what it is learned to classify or predict outcomes for fresh data.

2 .Unsupervised Learning: A type of machine learning uses algorithms to analyze unstructured data to find hidden patterns, structures, and relationships without human guidance.

3. Semi-Supervised Learning : A machine learning technique that uses a combination of both labeled and unlabeled data for training models.

4. Reinforcement Learning : Reinforcement learning is when a computer learns by trial and error. It tries different actions, gets rewards for good choices or penalties for bad ones, and figures out the best way to behave over time.

5. Feature / Attribute : A measurable property or column of the data used as input. A feature is like a main part of something, and then an attribute is more like a detail that describes that part.

6. Parameter:  a parameter is internal variable that the model learns from data to make predictions.

7. Training Data: Training data is the initial dataset used to teach an AI model what to do. It uses this information to learn the rules and patterns needed for its task.

**8. Validation Set:**This is like a practice quiz you give the model during training. You don't train on it, but you use it to check how it's doing and tune your settings before the final exam.

**9. Test Set:**This is the final exam. You only use it once, at the very end, to see how good your model really is on totally new, unseen data. It gives you the real-world performance score.

**10. Overfitting:**This is when the model memorizes the training data perfectly, including all the noise and weird details. It aces the practice tests but then fails the real exam because it didn't learn the general concepts—it just memorized the answers.

**11. Underfitting:** This is when the model is too simple and doesn't even learn the training data properly. It's like a student who didn't study enough—they fail the practice questions and the real exam.

**12. Bias–Variance Tradeoff:**It's this balancing act. You want a model that's not too simple (high bias, underfitting) and not too complex (high variance, overfitting). It's like finding the sweet spot where it learns the real patterns without getting distracted by the noise.

**13. Loss Function:**This is like a report card for a single training example. It tells the model how wrong its prediction was for just one piece of data. "For this one student, you were this much off."

**14. Cost Function:**This is the average report card for the entire class (the whole training set). It tells the model how wrong it is on average across all the examples. This is the main score the model tries to minimize.

**15. Gradient Descent:**This is the model's way of finding the lowest point in a valley (the lowest error). It looks around, figures out which direction is downhill, and takes a small step that way. It repeats this until it finds the bottom.

**16. Learning Rate:**This is the size of the step the model takes down the hill. Too big a step (high learning rate) and you might overshoot the bottom. Too small a step (low learning rate) and it takes forever to get there.

**17. Epoch:**One full pass through the entire training dataset. It's like the model reading a textbook from cover to cover once. You usually need multiple epochs (rereading the book) to learn well.

**18. Cross-validation:**A fancy way of testing. Instead of one train/test split, you split your data into several parts. You train on some and validate on others, rotating through all the parts to make sure your model's performance is consistent and not just lucky.

**19. Batch Size:**The number of training examples you look at before updating the model. A small batch is like studying with flashcards one by one. A large batch is like reviewing a whole chapter before testing yourself.

**20. Regularization:**This is basically a set of tricks to prevent the model from overfitting. It's like forcing it to use a simpler explanation, which often works better on new data. You're penalizing complexity.

**21. Linear Regression:**A simple model that finds the best-fit straight line through your data. It's used to predict a continuous number, like predicting house prices based on their size.

**22. Logistic Regression:**Despite the name, it's used for classification (putting things into categories). It finds an S-shaped curve to predict the probability of something belonging to a certain class (like spam or not spam).

**23. Decision Tree:**A model that makes predictions by asking a series of yes/no questions, like a game of "20 Questions." It splits the data based on features (e.g., "Is the age > 30?") until it reaches a conclusion.

**24. Random Forest:**A team of many decision trees. Instead of relying on one tree, it asks a whole bunch of them and then goes with the majority vote. This makes it much more robust and accurate than a single tree.

**25. Support Vector Machine (SVM):**A classifier that tries to find the best possible boundary (a line or a plane) to separate different classes in your data. It wants the widest possible "street" between the classes.

**26. K-Nearest Neighbors (KNN)**  
It classifies a new data point by looking at the 'K' closest labeled examples around it and letting them vote on what it should be. Like judging a new student by the friends they hang out with.

**27. Naive Bayes**  
A simple classifier that assumes all the features contribute independently to the probability of an outcome. It's "naive" because this assumption is rarely perfectly true in real life, but it often works surprisingly well anyway.

**28. Principal Component Analysis (PCA)**  
A technique to simplify your data by finding the most important "directions" of variation and projecting the data onto them. It's like summarizing a detailed face photo using just a few key features (distance between eyes, jawline shape) instead of every single pixel.

**29. Clustering**  
The goal of unsupervised learning where you group similar data points together without using pre-defined labels. Finding natural "clumps" or "gangs" in your data.

**30. Dimensionality Reduction**  
The process of reducing the number of random variables (features) in your data. PCA is one way to do this. It's like compressing a high-resolution image into a smaller file while trying to keep the main contents recognizable.

**31. Perceptron**  
The simplest possible type of neural network. A single "neuron" that takes inputs, weighs them, and produces a single output. It's the building block for everything more complex.

**32. Activation Function**  
A function in a neural network that decides whether a neuron should be "activated" (fired) or not. It introduces non-linearity, allowing the network to learn complex patterns, not just straight lines.

**33. Backpropagation**  
The "learning" algorithm for neural networks. It calculates how much each neuron contributed to the final error and then adjusts all the weights backwards through the network to do better next time.

**34. Convolutional Neural Network (CNN)**  
A type of neural network super good at processing images. It uses "filters" to scan across an image and detect patterns like edges, shapes, and eventually complex objects.

**35. Recurrent Neural Network (RNN)**  
A neural network designed for sequential data (like time series or text). It has a "memory" of previous inputs, which helps it understand context.

**36. Long Short-Term Memory (LSTM)**  
A smarter, more complex type of RNN. It has a better-designed "memory" that can learn which past information is important to keep and which to forget, making it much more effective for long sequences.

**37. Transformer**  
A modern neural network architecture that relies entirely on an "Attention Mechanism" to process data. It's the core technology behind large language models like GPT and has largely replaced RNNs for many tasks because it's more efficient and powerful.

**38. Attention Mechanism**  
A technique that allows a model to focus on the most relevant parts of the input when making a prediction. Like when translating a sentence, it "pays attention" to the specific words in the original sentence that are most relevant to the word it's currently writing.

**39. Dropout**  
A regularization technique used during training where random neurons are temporarily "ignored." This prevents the network from becoming too reliant on any single neuron and forces it to learn more robust features.

**40. Gradient Vanishing Problem**  
A common issue in deep networks where the gradients (error signals) become incredibly small as they are propagated back to the early layers. This means the early layers learn very slowly or stop learning altogether.

**41. Transfer Learning**  
Taking a model that's already been trained on a large, general dataset and using it as a starting point for a new, specific task. It's like a doctor who becomes a specialist—they don't start from scratch, they build on their general medical knowledge.

**42. Fine-Tuning**  
The second step of transfer learning. After you take a pre-trained model, you further train (or "tune") it on your own smaller, specific dataset to make it an expert on your particular problem.

**43. Hyperparameter Optimization**  
The process of finding the best settings (hyperparameters) for your model before training even starts. Things like the learning rate or number of layers. It's like tuning a race car's engine for peak performance before the race.

**44. Specificity**  
A metric that measures how good your model is at correctly identifying the negative cases. It's the "True Negative Rate." For a disease test, it's the percentage of healthy people correctly identified as healthy.

**45. Confusion Matrix**  
A table that summarizes how well your classification model performed. It breaks down the predictions into four categories: True Positives, True Negatives, False Positives, and False Negatives. It's the source for calculating most other metrics.

**46. Precision**  
Out of all the instances the model predicted as positive, how many were actually positive? It measures the model's accuracy when it makes a positive prediction. High precision means you can trust its "yes."

**47. F1 Score**  
A single metric that combines both Precision and Recall into one number. It's the "harmonic mean" of the two and is useful when you need a balance between them and your classes are imbalanced.

**48. ROC Curve**  
A graph that shows the performance of a classification model at all classification thresholds. It plots the True Positive Rate against the False Positive Rate.

**49. AUC (Area Under the Curve)**  
The area under the ROC Curve. It gives you a single number that summarizes the model's ability to distinguish between classes. An AUC of 1 is a perfect model; 0.5 is no better than random guessing.