ML 1: Uber Price Prediction (Linear Regression)

1. **What is the objective of this Linear Regression model?** To find the best-fit line that predicts continuous values (fare) based on independent variables.
2. **What are the independent and dependent variables here?** Independent: Distance, Passenger Count; Dependent: Fare Amount.
3. **Why do we need train_test_split?** To evaluate how well the model generalizes to new, unseen data.
4. **What is the default split ratio if not specified?** Usually 0.25 (25%) for testing in sklearn, but it's best practice to specify it (e.g., test_size=0.2).
5. **What does the .fit() method do?** It trains the model by calculating the optimal coefficients (weights) for the features.
6. **What does the .predict() method do?** It uses the trained coefficients to estimate the target value for new input data.
7. **What is an outlier in this dataset?** A ride with an unusually high fare for a short distance, or a negative fare.
8. **How does Linear Regression handle outliers?** Poorly; it is sensitive to them, and they can significantly skew the best-fit line.
9. **What is Correlation?** A measure (between -1 and 1) of how strongly two variables are related linearly.
10. **Does correlation imply causation?** No. Just because distance correlates with fare doesn't mean distance is the *only* cause of fare changes (e.g., traffic).
11. **What is the equation of a simple linear regression line?** $y = mx + c$ (or $y = b_0 + b_1x_1$).
12. **What do m and c represent?** m is the coefficient (slope/weight), and c is the intercept (bias).
13. **What is Mean Absolute Error (MAE)?** The average of the absolute differences between predicted and actual values.
14. **Why might you use MAE instead of RMSE?** MAE is less sensitive to outliers than RMSE.
15. **What is R-Squared (R^2) usually between?** Typically between 0 and 1 (though it can be negative if the model is worse than a simple horizontal line).

16. **What does an R^2 of 0.8 mean?** The model explains 80% of the variance in the target variable.
17. **What is Adjusted R^2?** A modified version of R^2 that adjusts for the number of predictors in the model; useful when comparing models with different numbers of features.
18. **What is Feature Engineering in this context?** Creating new meaningful features, like calculating the exact distance from latitude/longitude coordinates.
19. **What is "Geodesic Distance"?** The shortest path between two points on a curved surface (like the Earth), more accurate than straight-line Euclidean distance for maps.
20. **What happens if your data has Multicollinearity?** Independent variables are highly correlated with each other, making it hard to determine individual feature importance.
21. **What is "Underfitting" in this model?** When the model is too simple (e.g., a straight line for curved data) and performs poorly on both training and test data.

ML 2: Email Spam Classification (Naive Bayes/NLP)

1. **Is this Supervised or Unsupervised learning?** Supervised, because the email data comes with 'Spam' or 'Ham' labels.
2. **What is standard text preprocessing?** Lowercasing, removing punctuation, removing stopwords, and stemming/lemmatization.
3. **What is Tokenization?** Splitting a sentence or document into individual words or terms (tokens).
4. **What is a "Bag of Words" model?** Representing text by counting word frequencies, ignoring grammar and word order.
5. **What is the main disadvantage of Bag of Words?** It loses the context and meaning derived from the sequence of words.
6. **What does Naive Bayes assume?** It "naively" assumes that the presence of a particular word in a class is unrelated to the presence of any other word.
7. **What is Bayes' Theorem used for here?** To calculate the probability that an email is spam *given* the words it contains.

8. **What is a 'Prior Probability'?** The initial probability of an email being spam before looking at its content (e.g., overall 20% of all emails are spam).
9. **What is 'Likelihood' in Naive Bayes?** The probability of finding specific words *if* we already know the email is spam.
10. **Why do we use Laplace Smoothing in Naive Bayes?** To handle the problem of zero probability when a new word appears in the test data that wasn't in the training data.
11. **What is the difference between CountVectorizer and TfidfVectorizer?** CountVectorizer just counts word frequency; TfidfVectorizer weighs words by how unique they are to a specific document.
12. **What does a high TF-IDF score mean for a word?** The word is frequent in the current document but rare in other documents (likely a keyword).
13. **What is a Support Vector Machine (SVM) also good for?** It is effective in high-dimensional spaces, which is common in text classification.
14. **What is a "Kernel" in SVM?** A function that transforms data into a higher dimension to make it linearly separable.
15. **What is Accuracy?** The ratio of correctly predicted observations to total observations.
16. **What is Precision?** Out of all emails *predicted* as spam, how many were *actually* spam? (Important if you don't want to miss important emails).
17. **What is Recall (Sensitivity)?** Out of all *actual* spam emails, how many did the model correctly catch?
18. **What is the F1 Score?** The harmonic mean of Precision and Recall; good when you need a balance between the two.
19. **Why might Accuracy be misleading here?** If 99% of emails are Ham, a model that always predicts Ham has 99% accuracy but is useless.
20. **What is an N-gram?** A contiguous sequence of *n* items from a given text (e.g., "credit card" is a 2-gram or bigram).

ML 3: Bank Customer Churn (ANN)

1. **What are the three main types of layers in an ANN?** Input layer, Hidden layer(s), and Output layer.

2. **What is a Perceptron?** The simplest type of artificial neuron; a linear binary classifier.
3. **What is the role of 'Weights' in ANN?** They determine the importance/strength of the connection between neurons.
4. **What is the role of 'Bias' in ANN?** It allows the activation function to be shifted left or right, helping the model fit data better.
5. **Why do we need non-linear Activation Functions?** Without them, a neural network, no matter how deep, would just be a single linear regression model.
6. **Why is ReLU (Rectified Linear Unit) commonly used in hidden layers?** It is computationally efficient and helps avoid the "vanishing gradient" problem.
7. **Why use the Sigmoid function for the output layer here?** Because it squashes the output between 0 and 1, perfect for binary probability (Churn vs No Churn).
8. **What is 'Forward Propagation'?** Passing input data through the network layers to get a prediction.
9. **What is the 'Loss Function' used for binary classification ANN?** Binary Cross-Entropy (also called Log Loss).
10. **What does the 'Adam' optimizer do?** It's an adaptive learning rate optimization algorithm that is generally faster and more efficient than standard Gradient Descent.
11. **What is an Epoch?** One full pass of the entire training dataset through the network.
12. **What is Batch Size?** The number of training examples used in one iteration to update the weights.
13. **What is Overfitting in ANN?** When the network memorizes the training data noise and fails on new customers.
14. **How does 'Dropout' help prevent overfitting?** It randomly ignores (drops) some neurons during training to prevent over-reliance on specific paths.
15. **What is One-Hot Encoding?** Converting categorical variables (like 'France', 'Spain') into binary columns (0s and 1s).
16. **Why is Feature Scaling critical for ANN?** If one feature has a huge range (e.g., salary), it will dominate the weight updates and slow down training.
17. **What is the 'Vanishing Gradient Problem'?** When gradients become so small during backpropagation that early layers stop learning.

18. **What is 'Early Stopping'?** Stopping training when the validation performance stops improving, even if epochs aren't finished.
19. **Can an ANN have zero hidden layers?** Yes, it effectively becomes a Logistic Regression model.
20. **What is a 'Dense' layer in Keras/TensorFlow?** A fully connected layer where every neuron is connected to every neuron in the previous layer.

ML 4: Gradient Descent Algorithm

1. **In simple terms, what is Gradient Descent doing?** It's trying to find the lowest point (minimum error) in a landscape by taking downhill steps.
2. **What is the 'Gradient'?** The vector of partial derivatives; it points in the direction of the steepest increase of the function.
3. **Why do we subtract the gradient during updates?** Because we want to move *opposite* to the increase (we want to go downhill to minimize error).
4. **What is a convex function?** A bowl-shaped function with only one global minimum (easiest for Gradient Descent).
5. **What is a non-convex function?** A wavy function with multiple hills and valleys (many local minima).
6. **What is a Local Minimum?** A point that is lower than its immediate neighbors but not the lowest point overall.
7. **What is a Saddle Point?** A point where the gradient is zero but it's neither a minimum nor a maximum (flat area).
8. **What happens if the Learning Rate is too small?** The model will take forever to converge (too many tiny steps).
9. **What is Batch Gradient Descent?** It calculates the error for the *entire* dataset before taking one step (slow for big data).
10. **What is Stochastic Gradient Descent (SGD)?** It takes a step after every *single* data point (fast but very noisy/jittery updates).
11. **What is Mini-Batch Gradient Descent?** A compromise; it takes a step after a small batch of data (e.g., 32 or 64 examples).

12. **What is 'Momentum' in Gradient Descent?** A technique that helps accelerate SGD in the relevant direction and dampens oscillations.
13. **What does it mean if the cost function is increasing over time?** Your learning rate is likely too high, causing the algorithm to overshoot and diverge.
14. **What is the stopping criterion for Gradient Descent?** Usually when the change in cost between iterations is very small (convergence) or a maximum number of iterations is reached.
15. **Can Gradient Descent be used for non-ML problems?** Yes, it's a general mathematical optimization algorithm.
16. **How does Feature Scaling affect Gradient Descent?** It makes the cost function "bowl" rounder rather than elongated, allowing faster, more direct convergence.
17. **What is the 'Cost Function' for Linear Regression that GD minimizes?** Usually Mean Squared Error (MSE).
18. **What is the difference between a parameter and a hyperparameter?** Parameters (weights) are learned by GD; hyperparameters (learning rate) are set by you before training.
19. **What is Learning Rate Decay?** Gradually reducing the learning rate over time to take smaller steps as you get closer to the minimum.
20. **Why do we initialize weights randomly instead of all zeros?** To break symmetry; if all weights are zero, all neurons learn the same thing.

ML 5: KNN Algorithm (Diabetes)

1. **Why is KNN called "Non-Parametric"?** It doesn't make any assumptions about the underlying data distribution (like believing data is normally distributed).
2. **Why is KNN called a "Lazy Learner"?** It doesn't do any work during training; it waits until a query is made to perform calculations.
3. **What is the 'Curse of Dimensionality' in KNN?** As the number of features grows, "closeness" becomes less meaningful, and you need vastly more data.
4. **How do you choose the optimal 'K'?** Usually by trial and error using techniques like cross-validation and picking the K with the lowest error.

5. **Why is an odd number usually preferred for 'K'?** To avoid tied votes in binary classification (e.g., 2 votes for Diabetes, 2 votes for No Diabetes).
6. **What is Euclidean Distance?** Straight-line distance between two points: $\sqrt{\sum(x_i - y_i)^2}$.
7. **What is Manhattan Distance?** The sum of absolute differences (grid-like path): $\sum|x_i - y_i|$.
8. **What is Minkowski Distance?** A generalized distance metric; it becomes Manhattan when p=1 and Euclidean when p=2.
9. **How does Feature Scaling affect KNN?** Drastically; without it, features with large ranges (like Insulin 0-800) dominate features with small ranges (like Age 20-80).
10. **Can KNN be used for regression?** Yes, instead of voting, it averages the values of the K nearest neighbors.
11. **What happens if K = N (total number of data points)?** The model will just predict the majority class of the entire dataset for everything.
12. **What happens if K = 1?** The model is very sensitive to noise (overfitting), predicting based only on the single closest (potentially noisy) point.
13. **What is 'Weighted' KNN?** Closer neighbors are given more influence (higher voting weight) than farther neighbors.
14. **Is KNN sensitive to outliers?** Yes, especially with small K values, an outlier can determine the prediction for nearby points.
15. **What is the main computational drawback of KNN?** It is very slow at prediction time, especially with large datasets, because it must calculate distances to *all* points.
16. **How can you speed up KNN?** Using data structures like KD-Trees or Ball Trees instead of brute-force distance calculation.
17. **Is KNN good for imbalanced data?** Not naturally; the majority class tends to dominate the nearest neighbors.
18. **What is 'Cross-Validation'?** Splitting data into multiple parts and training/testing on different combinations to get a robust estimate of performance.
19. **What does the 'Diabetes Pedigree Function' feature likely represent?** A score indicating the likelihood of diabetes based on family history.

20. **In medical testing, what is worse: False Positive or False Negative?** Usually False Negative (missing a patient who actually has diabetes), though False Positive leads to unnecessary stress/tests.

ML 6: K-Means Clustering (Sales Data)

1. **Is this Supervised or Unsupervised?** Unsupervised; we don't have labels telling us which customer belongs to which segment.
2. **What is a 'Centroid'?** The center point of a cluster, calculated as the average (mean) of all points currently assigned to that cluster.
3. **How does the K-Means algorithm start?** By randomly selecting 'K' initial centroids.
4. **What are the two main steps repeated in K-Means?** Assignment (assign points to nearest centroid) and Update (recalculate centroids based on new assignments).
5. **When does K-Means stop (converge)?** When centroids no longer move significantly, or points stop changing clusters.
6. **What is WCSS (Within-Cluster Sum of Squares)?** The sum of the squared distances between each point and its assigned centroid (measures cluster compactness).
7. **What is the 'Elbow Method'?** Plotting WCSS against K; the "elbow" is the point where adding more clusters doesn't significantly improve WCSS.
8. **What is the 'Silhouette Score'?** A measure (-1 to 1) of how similar a point is to its own cluster compared to other clusters (higher is better).
9. **What is the main limitation of K-Means regarding cluster shape?** It assumes clusters are spherical and roughly the same size; it fails on elongated or irregular shapes.
10. **How does K-Means handle outliers?** Poorly; outliers can pull the centroid far away from the real cluster center.
11. **What is K-Means++?** A smarter initialization technique that chooses initial centroids far apart from each other to speed up convergence.
12. **Is K-Means deterministic?** No, different random initializations can lead to different final clusters.

13. **What is the difference between Clustering and Classification?** Classification predicts predefined labels (supervised); Clustering discovers grouping structures in data (unsupervised).
14. **What is 'Hard' vs 'Soft' clustering?** Hard: a point belongs to exactly one cluster. Soft (Fuzzy): a point has a probability of belonging to each cluster.
15. **Why do we need to scale data for K-Means?** It relies entirely on distance; if one feature (e.g., Annual Income) has a huge scale, it will dominate the distance calculations.
16. **What real-world business value does this lab provide?** Customer Segmentation—allowing businesses to target marketing specifically to "high spenders" vs "budget shoppers".
17. **What is Agglomerative Hierarchical Clustering?** An alternative where every point starts as its own cluster, and they are iteratively merged.
18. **What is a Dendrogram?** A tree-like diagram used to visualize the steps of hierarchical clustering.
19. **What if your data has categorical variables?** Standard K-Means doesn't work well; you might need K-Modes or to encode them appropriately.
20. **How do you interpret the final clusters?** By looking at the average values of features (e.g., "Cluster 1 has low age but high spending score").