# Logistic Regression

1. Describe the squashing effect in logistic regression and how it relates to the sigmoid function.  
  
 The squashing effect in logistic regression refers to the transformation of a linear combination of input features into a probability score between 0 and 1. This transformation is accomplished using the sigmoid function, defined as:  
  
 σ(z) = 1 / (1 + e^(-z))  
  
 where z is the linear combination of the input features and their corresponding weights. The sigmoid function maps any real-valued number into the (0, 1) interval, thus squashing the output to represent a probability.  
  
 2. Explain how the optimization problem in logistic regression is solved.  
  
 The optimization problem in logistic regression is typically solved by maximizing the log-likelihood function. The goal is to find the optimal parameters (weights) that maximize the likelihood of observing the given data under the logistic regression model.  
  
 - Log-Likelihood Function: For a dataset with n samples, where (x\_i, y\_i) are the input features and labels:  
  
 ℓ(θ) = Σ\_i=1 to n [y\_i \* log(h\_θ(x\_i)) + (1 - y\_i) \* log(1 - h\_θ(x\_i))]  
  
 where h\_θ(x\_i) = σ(θ^T \* x\_i) is the predicted probability for the i-th sample.  
  
 - Optimization Technique: Gradient descent or other iterative optimization methods are commonly used to find the parameters θ that maximize ℓ(θ).  
  
 3. What is the role of the log-likelihood function in logistic regression?  
  
 The log-likelihood function serves as the objective function to be maximized during the training of a logistic regression model. It measures how well the model predicts the observed data by comparing the predicted probabilities h\_θ(x\_i) with the actual labels y\_i. Maximizing the log-likelihood function helps in finding the parameters θ that best fit the data, thereby optimizing the predictive performance of the logistic regression model.  
  
 4. Logistic Regression vs. Linear Regression  
  
 - Use Cases:  
 - Logistic Regression: Used for binary classification problems where the outcome is categorical (e.g., yes/no, true/false).  
 - Linear Regression: Used for predicting continuous numeric values (e.g., predicting house prices).  
  
 - Assumptions:  
 - Logistic Regression: Assumes a linear relationship between the log-odds of the dependent variable and the independent variables. Assumes independence of observations.  
 - Linear Regression: Assumes a linear relationship between the dependent and independent variables. Assumes homoscedasticity (constant variance of errors), independence of errors, and normally distributed errors.  
  
 - Interpretations:  
 - Logistic Regression: Coefficients represent the log-odds change for a one-unit change in the predictor.  
 - Linear Regression: Coefficients represent the change in the dependent variable for a one-unit change in the predictor.  
  
 5. Applications of Logistic Regression in Real-World Scenarios  
  
 a. Medical Diagnosis:  
 Logistic regression is widely used in medical fields for predicting the probability of a disease based on patient characteristics. For example, predicting the likelihood of heart disease based on factors like age, blood pressure, and cholesterol levels.  
  
 b. Marketing:  
 In marketing, logistic regression helps in customer segmentation and targeting. It can predict the probability of a customer buying a product based on demographic data, purchase history, and online behavior.

# Naive Bayes Classifier

6. Define Naive Bayes Feature importance and interpretability.  
  
 Naive Bayes classifiers assume that features are conditionally independent given the class label. Feature importance in Naive Bayes can be inferred from the conditional probabilities P(x\_i | y), which indicate the strength of association between each feature x\_i and the class y.  
  
 7. Write the formula for the Naive Bayes classifier and explain each term.  
  
 The Naive Bayes classifier formula is:  
  
 P(y | X) = (P(y) \* P(X | y)) / P(X)  
  
 - P(y | X): Posterior probability of class y given features X.  
 - P(y): Prior probability of class y.  
 - P(X | y): Likelihood of features X given class y.  
 - P(X): Evidence or marginal likelihood of features X.  
  
 Naive Bayes calculates these probabilities assuming independence among features given the class, simplifying the computation.  
  
 12. Discuss the impact of outliers on the performance of the Naive Bayes classifier.  
  
 Outliers can significantly affect the performance of Naive Bayes because the classifier relies on the assumption of feature independence. Outliers can distort the feature distributions, leading to incorrect probability estimates and potentially reducing classification accuracy.  
  
 24. Given the following dataset, use the Naive Bayes algorithm to predict whether to play tennis if the weather is sunny and the temperature is mild. Show all the calculations.  
  
 Dataset:  
 Weather Temperature Play Tennis  
 Sunny Hot No  
 Overcast Hot Yes  
 Rainy Mild Yes  
 Sunny Cool Yes  
 Rainy Cool No  
  
 To predict whether to play tennis when weather is sunny and temperature is mild:  
 - Calculate prior probabilities P(Yes) and P(No).  
 - Calculate likelihoods P(Sunny | Yes), P(Mild | Yes), P(Sunny | No), P(Mild | No).  
 - Compute posteriors P(Yes | Sunny, Mild) and P(No | Sunny, Mild).  
  
 26. Discuss the impact of outliers on the performance of the Naive Bayes classifier (Repeated)  
  
 Outliers can distort feature distributions, violating the independence assumption in Naive Bayes. This can lead to incorrect probability estimates and reduced classification accuracy.

# Data Visualization in Matplotlib and Seaborn

1. Describe the difference between xlim() and ylim() in Matplotlib.  
  
 - xlim(): Sets or gets the x-axis limits of the current plot.  
 - ylim(): Sets or gets the y-axis limits of the current plot.  
  
 11. What is a whisker plot, and how does it relate to a box plot?  
  
 A whisker plot, also known as a box plot, displays the distribution of data based on a five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. Whiskers extend from the box to show the range of the data within 1.5 times the interquartile range (IQR) from Q1 and Q3.  
  
 15. Define what subplots and KDE are in the context of data visualization  
  
 - Subplots: Multiple plots within a single figure, allowing for comparison of different datasets or views of the same dataset in one visual context.  
 - KDE (Kernel Density Estimation): A non-parametric way to estimate the probability density function of a continuous random variable. It smooths data points to create a continuous curve representing the data distribution.  
  
 19. Illustrate the importance of data visualization in data analysis and explain about pairwise plot, violin plot, and palette in Seaborn.  
  
 Data visualization is crucial for:  
 - Understanding patterns and relationships in data.  
 - Communicating insights effectively.  
 - Identifying outliers and trends.  
  
 - Pairwise Plot: Shows pairwise relationships in a dataset, useful for identifying correlations and patterns.  
 - Violin Plot: Combines a box plot and a KDE plot, showing the distribution of data across different categories.  
 - Palette: A color scheme used in Seaborn to enhance visual appeal and readability of plots.  
  
 22. Explain how to create a KDE plot in Seaborn. Discuss the advantages of using KDE plots over histograms in certain scenarios. Provide a code example that demonstrates how to customize a KDE plot.  
  
 To create a KDE plot in Seaborn:  
  
 ```python  
 import seaborn as sns  
 import matplotlib.pyplot as plt  
  
 # Sample data  
 data = sns.load\_dataset("iris")  
  
 # KDE plot  
 sns.kdeplot(data['sepal\_length'], shade=True, color='r')  
  
 # Customization  
 plt.title('KDE Plot of Sepal Length')  
 plt.xlabel('Sepal Length')  
 plt.ylabel('Density')  
  
 plt.show()  
 ```  
  
 Advantages of KDE plots over histograms:  
 - Provides a smooth estimate of the data distribution.  
 - Less sensitive to the choice of bin width.  
 - Useful for visualizing continuous data distributions without discrete binning artifacts.