Logistic Regression

**1. Data Exploration:**

a. Load the dataset and perform exploratory data analysis (EDA).

b. Examine the features, their types, and summary statistics.

c. Create visualizations such as histograms, box plots, or pair plots to visualize the distributions and relationships between features.

Analyze any patterns or correlations observed in the data.

**2. Data Preprocessing:**

a. Handle missing values (e.g., imputation).

b. Encode categorical variables.

**3. Model Building:**

a. Build a logistic regression model using appropriate libraries (e.g., scikit-learn).

b. Train the model using the training data.

**4. Model Evaluation:**

a. Evaluate the performance of the model on the testing data using accuracy, precision, recall, F1-score, and ROC-AUC score.

Visualize the ROC curve.

**5. Interpretation:**

a. Interpret the coefficients of the logistic regression model.

b. Discuss the significance of features in predicting the target variable (survival probability in this case).

**6. Deployment with Streamlit:**

In this task, you will deploy your logistic regression model using Streamlit. The deployment can be done locally or online via Streamlit Share. Your task includes creating a Streamlit app in Python that involves loading your trained model and setting up user inputs for predictions.

(optional)For online deployment, use Streamlit Community Cloud, which supports deployment from GitHub repositories.

Detailed deployment instructions are available in the Streamlit Documentation.

<https://docs.streamlit.io/streamlit-community-cloud/deploy-your-app>

**Interview Questions:**

1. What is the difference between precision and recall?

Answer: **Precision** and **recall** are two important metrics used to evaluate the performance of a classification model, especially in binary classification. Both metrics are derived from the confusion matrix, which consists of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

**Precision**:

* Precision is the ratio of correctly predicted positive observations to the total predicted positives.
* It answers the question: "Of all the instances that were predicted as positive, how many were actually positive?"
* Formula: Precision =
* High precision indicates that the model has a low false positive rate.

**Recall** (also known as Sensitivity or True Positive Rate):

* Recall is the ratio of correctly predicted positive observations to all observations in the actual class.
* It answers the question: "Of all the instances that are actually positive, how many were correctly predicted as positive?"
* Formula: Precision =
* High recall indicates that the model has a low false negative rate.

**Key Difference**:

* Precision focuses on the quality of positive predictions (minimizing false positives), whereas recall focuses on the completeness of positive predictions (minimizing false negatives). In other words, precision is about correctness, while recall is about coverage.

2. What is cross-validation, and why is it important in binary classification?

Answer: **Cross-validation** is a technique used to assess the generalizability and robustness of a machine learning model. It involves partitioning the dataset into multiple subsets and training/testing the model multiple times to ensure that the model performs well on unseen data. The most common form of cross-validation is **k-fold cross-validation**.

**K-fold Cross-Validation**:

* The dataset is divided into kkk equally sized folds (subsets).
* The model is trained kkk times, each time using k−1k-1k−1 folds for training and the remaining fold for testing.
* The process is repeated kkk times, with each fold used exactly once as the testing set.
* The final performance metric is the average of the metrics from each fold.

**Importance of Binary Classification:**

1. **Prevents Overfitting**:
   * By training and testing the model on different subsets of the data, cross-validation helps to ensure that the model does not memorize the training data but rather learns to generalize to new, unseen data.
2. **Provides a More Reliable Estimate of Model Performance**:
   * It reduces the variability of the performance metric by averaging the results across multiple folds. This provides a more stable and reliable estimate of how the model will perform on unseen data compared to a single train-test split.
3. **Ensures Fair Evaluation**:
   * In binary classification, especially with imbalanced datasets, cross-validation ensures that both classes are adequately represented in both training and testing sets across multiple iterations, leading to a more fair and thorough evaluation.
4. **Helps in Hyperparameter Tuning**:
   * Cross-validation is often used in conjunction with grid search or random search to tune hyperparameters. It ensures that the selected hyperparameters generalize well to new data.

By using cross-validation, you can be more confident that your model is robust, and that the performance metrics are not overly optimistic or pessimistic due to the specific train-test split.