Data Splitting

- **Process:** Divide the dataset into training and testing subsets.
  - **Key Phrase:** `X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)`
  - **Callout:** 80% of data used for training, 20% for testing. `random_state` ensures reproducibility.

Model Initialization

- **Process:** Select and initialize the classification algorithm.
  - **Key Phrase:** `model = LogisticRegression(solver='liblinear', random_state=42)`
  - **Callout:** Logistic Regression model chosen for binary classification.

Model Training

- **Process:** Train the model using the prepared training data.
  - **Key Phrase:** `model.fit(X_train, y_train)`
  - **Callout:** The model learns patterns from the features and their corresponding target labels.

Model Prediction

- **Process:** Generate predictions on the unseen test data.
  - **Key Phrase:** `y_pred = model.predict(X_test)`
  - **Callout:** Model forecasts landing outcomes (0 or 1) for the test set.

Model Evaluation

- **Process:** Assess the model's performance using various metrics.
  - **Key Phrase (Accuracy):** `accuracy = accuracy_score(y_test, y_pred)`
  - **Key Phrase (Confusion Matrix):** `conf_matrix = confusion_matrix(y_test, y_pred)`
  - **Key Phrase (Classification Report):** `print(classification_report(y_test, y_pred))`
  - **Callout:** Metrics provide insight into correct/incorrect predictions, precision, recall, and F1-score.