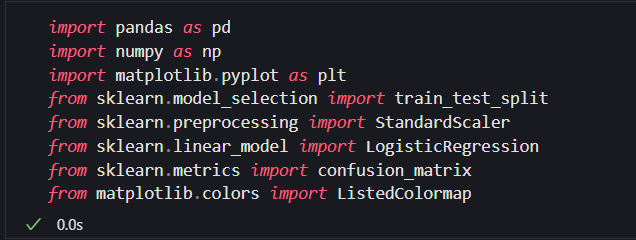
LOGISTIC REGRESSION MODEL

**Discription:**This document demonstrates the process of visualizing a logistic regression model using the training data and testing data from the Car\_Purchase\_Prediction dataset. The purpose is to understand the relationship between the Age, Salary, and the Purchased variable by fitting a logistic model and visualizing the decision boundary.

1. **Importing Libraries**



**Explanation:**

This section imports the necessary libraries:

• pandas for data manipulation and analysis.

• numpy for numerical operations.

• train\_test\_split from sklearn for splitting the dataset.

• StandardScaler from sklearn for feature scaling.

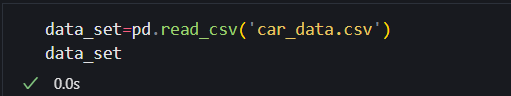
• LogisticRegression from sklearn for building the logistic regression model.

• confusion\_matrix from sklearn for evaluating model performance.

• ListedColormap from matplotlib.colors for creating custom color maps.

• matplotlib.pyplot for plotting graphs.

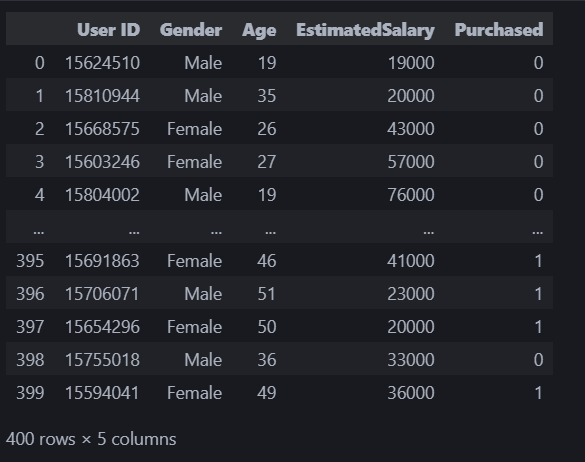
1. **Loading the Dataset**



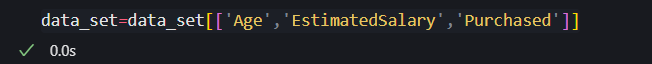
**Explanation:**

To load a dataset using pandas, you can use the pd.read\_csv() function. This function reads a CSV (Comma-Separated Values) file and creates a DataFrame, which is a 2-dimensional labeled data structure with columns of potentially different types.

**Output:**



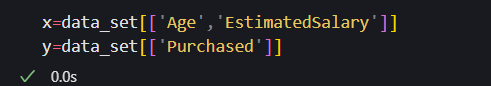
1. **Selecting Relevant Columns**



**Explanation:**

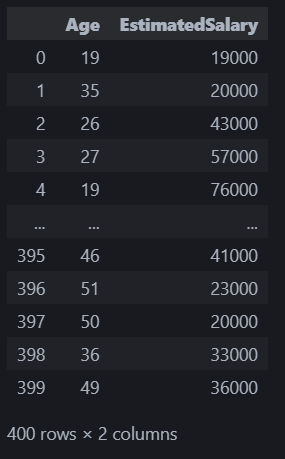
This filters the dataset to include only the columns "Age", "EstimatedSalary", and "Purchased," which are the features and target variable used for this analysis.

1. **Defining Features and Target Variables**

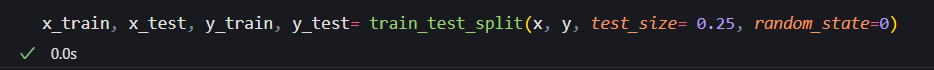


**Explanation:**  
Here, x represents the feature variables (Age and EstimatedSalary), and y represents the target variable (Purchased).

Output:



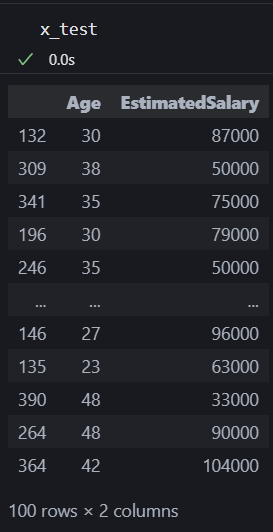
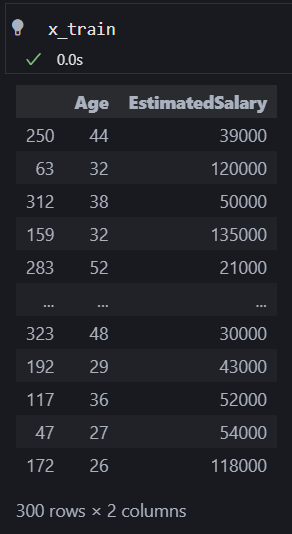
1. **Splitting the Dataset**



**Explanation:**

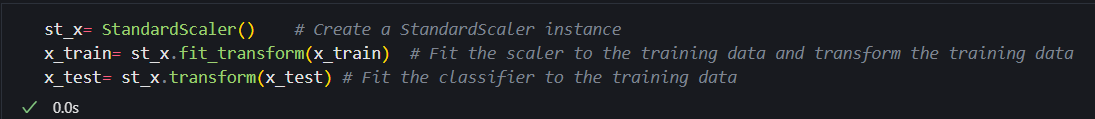
The dataset is split into training and test sets. 75% of the data is used for training, and 25% is used for testing.

**Output:**



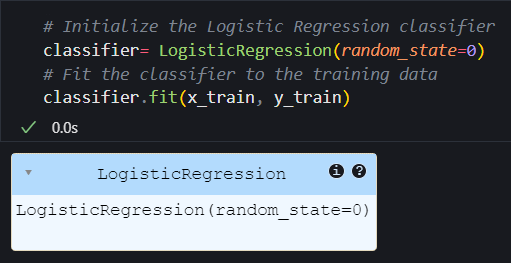


1. **Data Preprocessing:**



**Explanation:**  
The StandardScaler standardizes the features by removing the mean and scaling to unit variance. The fit\_transform method is applied to the training data to compute the scaling parameters, and transform is used to apply the same scaling to the test data.

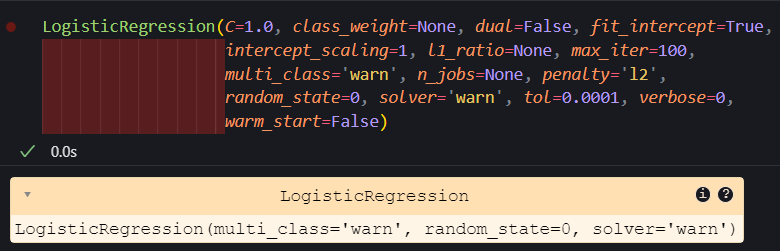
1. **Data Fitting:**



**Explanation:**  
A LogisticRegression model is created and trained on the standardized training data. The model learns to predict the target variable based on the features.

1. **Predicting and Evaluating the Model:**

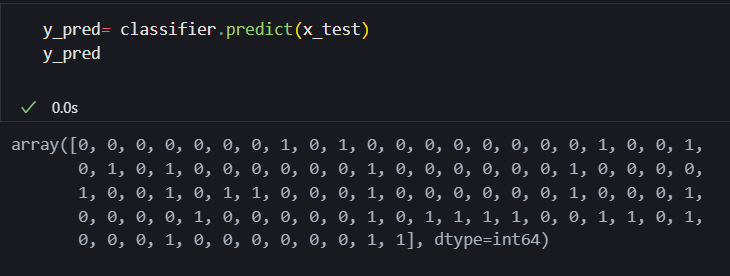
**LogisticRegression Initialization:**



**Default Parameters**:

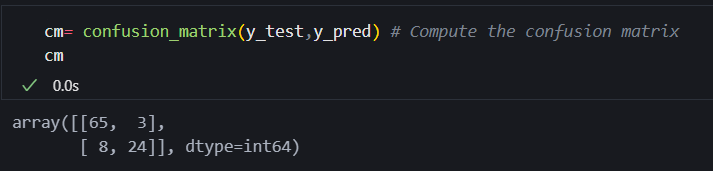
* **C=1.0**: Regularization strength.
* **class\_weight=None**: No weights applied to classes.
* **dual=False**: Dual formulation not used.
* **fit\_intercept=True**: Includes an intercept in the model.
* **intercept\_scaling=1**: Scaling of the intercept term.
* **l1\_ratio=None**: No elastic net regularization.
* **max\_iter=100**: Maximum number of iterations.
* **multi\_class='warn'**: Behavior for multi-class classification.
* **n\_jobs=None**: Number of parallel jobs.
* **penalty='l2'**: L2 regularization used.
* **random\_state=0**: Seed for random number generator.
* **solver='warn'**: Solver for optimization.
* **tol=0.0001**: Tolerance for stopping criteria.
* **verbose=0**: No verbose output.
* **warm\_start=False**: Start with the previous solution.

**Predicting and Evaluating the Model:**



**Explanation**:  
The trained model is used to predict outcomes on the test data. The confusion\_matrix function evaluates the model's performance by comparing predicted values (y\_pred) with actual values (y\_test).

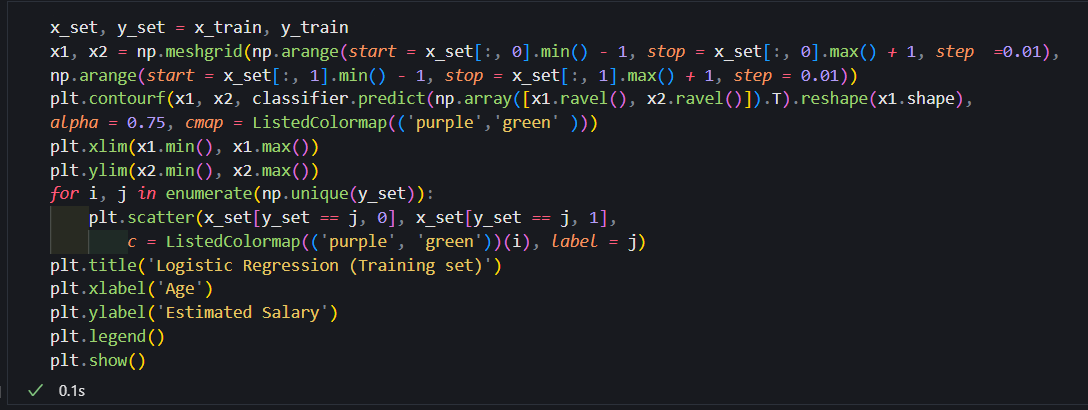
**Test Accuracy of the result**



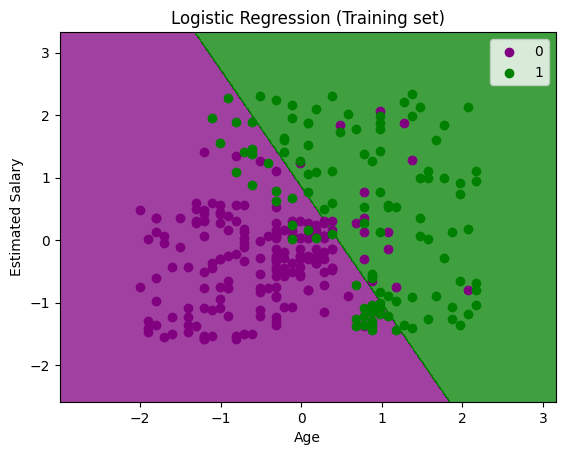
**Explanation:**

The code cm = confusion\_matrix(y\_test, y\_pred) computes the confusion matrix, which compares the true labels (y\_test) with the predicted labels (y\_pred). The confusion matrix summarizes the performance of the model by showing the counts of true positives, true negatives, false positives, and false negatives, helping to evaluate its accuracy and effectiveness.

1. **Visualizing the Training Set Results:**

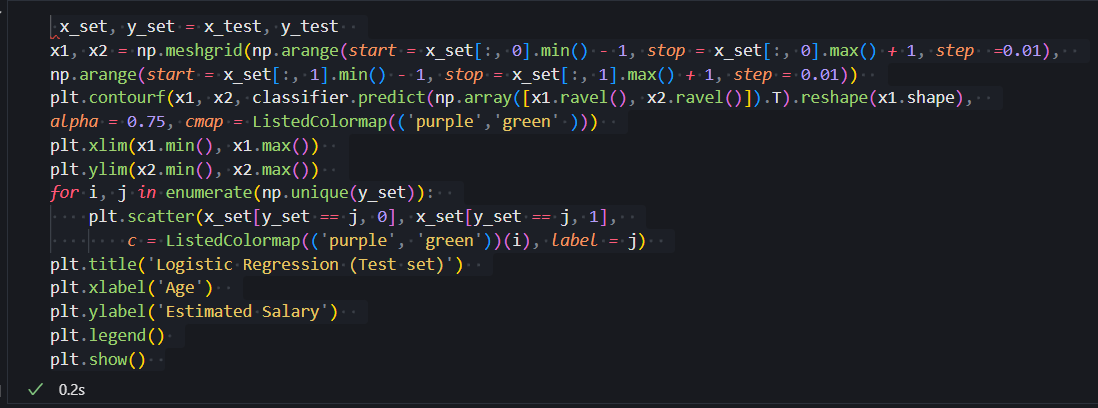


**Output:**



**Explanation**:  
This code visualizes the decision boundary of the logistic regression model trained on the training set. It creates a mesh grid, predicts the class for each point in the grid, and plots the decision boundary along with the training data points.

**10. Visualizing the Test Set Results**



Output:



**Explanation**:  
This code visualizes the decision boundary of the logistic regression model on the test set. It creates a mesh grid, predicts the class for each point in the grid, and plots the decision boundary along with the test data points.