

A PYTHON PROGRAM TO IMPLEMENT THE LOGISTIC MODEL

Aim:

To implement a Python program for the logistic model using the.

Algorithm:

Step 1: Import Necessary Libraries:

- pandas for data manipulation
- sklearn.model_selection for train-test split
- sklearn.preprocessing for data preprocessing
- sklearn.linear_model for logistic regression
- matplotlib.pyplot for plotting

Step 2: Read the Dataset:

- Use pandas to read the dataset.csv dataset into a DataFrame.

Step 3: Preprocess the Data:

- Select the relevant columns for the analysis (e.g., 'Age', 'EstimatedSalary', 'Purchased').
- Encode categorical variables if necessary (e.g., using LabelEncoder or OneHotEncoder).
- Split the data into features (X) and the target variable (y).

Step 4: Split the Data:

- Split the dataset into training and testing sets using

train_test_split. Step 5: Feature Scaling:

- Standardize the features using StandardScaler to ensure they have the same scale. Step 6: Create and Train the Model:

- Create a logistic regression model using LogisticRegression from sklearn.linear_model.

- Train the model on the training data using the fit method.
 - Create a function named “Sigmoid ()” which will define the sigmoid values using the formula $(1/(1+e^{-z}))$ and return the computed value.
 - Create a function named “initialize()” which will initialize the values with zeroes and assign the value to the “weights” variable, initialize with ones and assign the value to the variable “x”, and return both “x” and “weights”.
 - Create a function named “fit” which will be used to plot the graph according to the training data.
 - Create a predict function that will predict values according to the training model created using the fit function.

Step 7: Make Predictions:

- Use the trained model to make predictions on the test data using the predict method.
 - Use the “predict()” function to predict the values of the testing data and assign the value to the “y_pred” variable.
 - Use the “predict()” function to predict the values of the training data and assign the value to “y_train” variable.
 - Compute f1_score for both the training and testing data and assign the values to “f1_score_tr” and “f1_score_te” respectively

Step 8: Evaluate the Model:

- Calculate the accuracy of the model on the test data using the score method. ($\text{Accuracy} = (\text{tp} + \text{tn}) / (\text{tp} + \text{tn} + \text{fp} + \text{fn})$).
- Generate a confusion matrix and classification report to further evaluate the model's performance.

Step 9: Visualize the Results:

- Plot the decision boundary of the logistic regression model (optional).