**Lead scoring case study**

**Predicting the probability of lead conversion using Logistic Regression Model**

**Submitted by: Rafeek Ponnandy, Priyanka Chatterjee, Radhika Bansal**

Here’s a summary of the activities performed to build the final logistic regression model for predicting lead conversion:

1. **Data Inspection and Preparation:**
   * **Read the data set from a CSV file using pd.read\_csv('Leads.csv') and checked the first few rows with df.head().**
   * **Used df.info() to get column information (data types, missing values).**
   * **Checked the shape of the Data Frame with df.shape.**
   * **Obtained summary statistics using df.describe().**
   * **Handled missing values by dropping columns with over 30% missing data and replacing ‘Select’ values with NaN.**
   * **Removed columns with high missing values or low variability.**
   * **Imputed missing values in categorical columns by replacing them with ‘missing’.**
2. **Dummy Variables, Test-Train Split, and Feature Scaling:**
   * **Converted categorical variables into numerical format using pd.get\_dummies().**
   * **Split the data set into training and testing subsets (e.g., 70-30 or 80-20 split).**
   * **Normalized numerical features (e.g., Min-Max scaling or Z-score normalization).**
3. **Model Building:**
   * **Used Recursive Feature Elimination (RFE) to select relevant features.**
   * **Assessed multicollinearity using Variance Inflation Factor (VIF).**
   * **Selected important features: ‘Lead Origin’, ‘Lead Source’, ‘Do Not Email’, ‘Converted’, ‘TotalVisits’, ‘Total Time Spent on Website’, ‘Page Views Per Visit’, ‘Last Activity’, ‘Specialization’, ‘What is your current occupation’, ‘A free copy of Mastering The Interview’, and ‘Last Notable Activity’.**
4. **Model Evaluation:**
   * **Logistic regression model predicted probabilities for lead conversion.**
   * **Created a DataFrame with actual conversion flag and predicted probabilities.**
   * **Set an arbitrary cutoff (e.g., 50%) for predictions.**
   * **Overall model accuracy was approximately 80%.**
   * **Confusion matrix:**
     + **True negatives (TN): 1949**
     + **False positives (FP): 392**
     + **False negatives (FN): 539**
     + **True positives (TP): 1614**
   * **Sensitivity (Recall) was approximately 75%, specificity around 83%.**
5. **Finding the Optimal Cutoff Point:**
   * **Created an ROC curve (AUC = 0.87).**
   * **Optimal cutoff point: Around 42% for improved performance.**
   * **Sensitivity increased to 79.7%, specificity at 79%, precision at 77.7%.**
6. **Making Predictions on the Test Set:**
   * **Scaled test set features and selected relevant columns.**
   * **Predicted conversion probabilities.**
   * **Evaluated model performance:**
     + **Accuracy: 77%**
     + **Sensitivity: 79%**
     + **Specificity: 76%**
     + **Precision: 76%**
7. **Precision and Recall View:**
   * **Built a model using precision-recall trade-off.**
   * **Precision: ~80%, Recall: ~75%.**
   * **Optimal threshold (training set): 0.4.**
   * **Confusion matrix:**
     + **TN: 185**
     + **FP: 491**
     + **FN: 435**
     + **TP: 171**
8. **Prediction on the Test Set:**
   * **Predicted conversion probabilities on the test set.**
   * **Accuracy on the test set: ~78%.**
   * **Confusion matrix:**
     + **True negatives (TN): 770**
     + **False positives (FP): 218**
     + **False negatives (FN): 201**
     + **True positives (TP): 737**