Our Approach:

To build a logistic regression model to assign a lead score between 0 and 100 to each lead, the following steps can be taken:

- Data Preprocessing: Perform data exploration and cleaning as described in my previous answers.
- Feature Selection: Select the most relevant features that will be used to predict lead conversion. This can be done using techniques such as correlation analysis, chi-squared test, or mutual information.
- Data Splitting: Split the data into training and test sets in order to evaluate the model's performance on unseen data.
- Model Training: Train a logistic regression model using the selected features and the training data.
- Model Evaluation: Evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1-score.
- Hyperparameter tuning: Adjust the parameters of the model to optimize its performance.
- Lead Scoring: Assign a lead score to each lead using the trained model. The logistic regression model will output a probability of conversion for each lead, which can be mapped to a score between 0 and 100.

Solution Summary:

Step1: Reading and Understanding Data: Read and inspected the data.

Step2: Data Cleaning:

- First step to clean the dataset we chose was to drop the variables having unique values.
- Then, there were few columns with value 'Select' which means the leads did not choose any given option.
- We changed those values to Null values. c. We dropped the columns having NULL values greater than 40%.
- Next, we removed the imbalanced and redundant variables. This step also included imputing the missing values as and where required with median values in case of numerical variables and creation of new classification variables in case of categorical variables.
- The outliers were identified and removed. Also, in one column was having identical label in different cases (first letter small and capital respectively). We fixed this issue by converting the label with first letter in small case to upper case.

Step3: Data Transformation: Changed the binary variables into '0' and '1'

Step4: Dummy Variables Creation:

- We created dummy variables for the categorical variables.
- Removed all the repeated and redundant variables

Step5: Test Train Split:

- The next step was to divide the data set into test and train sections with a proportion of 70-30% values.

Step6: Feature Rescaling:

- We used the Min Max Scaling to scale the original numerical variables.
- The, we plot the a heatmap to check the correlations among the variables.
- Dropped the highly correlated dummy variables.

Step7: Model Building:

- Using the Recursive Feature Elimination, we went ahead and selected the 18 top important features.
- Using the statistics generated, we recursively tried looking at the P-values in order to select the most significant values that should be present and dropped the insignificant values.
- Finally, we arrived at the most significant variables.
- The VIF's for these variables were also found to be good.
- For our final model we checked the optimal probability cut off by finding points and checking the accuracy, sensitivity and specificity.
- We then plot the ROC curve for the features and the curve came out be pretty decent with an area coverage of 98% which further solidified the of the model.

Conclusion:
The model is accurate with below metric and makes sense to be used across the org

Train Data

Metrics	
Overall Accuracy	93%
Sensitivity	93.3%
Specificity	92.9%
Based on Confusion Matrix	
False Positive rate	7%
Positive Predicted Rate	89%
Negative Predicted rate	95.7%
Precision and Recall	
Precision	89%
Recall	93.3%

Test Data

Metrics	
Overall Accuracy	93.4%
Sensitivity	95.2%
Specificity	92.3%
Precision and Recall	
Precision	88%
Recall	95.2%