Summary Report

Reading and Understanding the Data

Initial data has 9240 records in leads.csv with 37 columns: 30 categorical and 7 numerical.

Data Clean up

- Replaced ‘Select’ with NaN.
- Dropped columns with one unique value or more than 40% missing values.
- Created new categories for variables with many classes.
- Replaced NaN with ‘others’ for Specialization

Visualizing Data and EDA

- Box Plot: TotalVisits, Total Time Spent on Website, Page Views Per Visit.
- Pair Plot: Numeric variables.
- Count Plot: Categorical variables with Converted.

Data Preparation

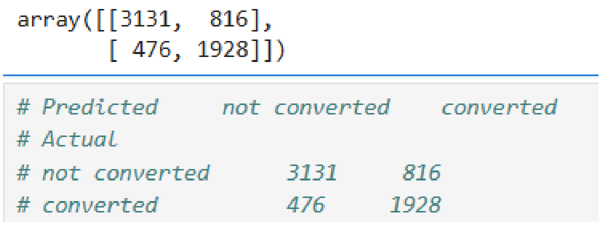
- Identified and removed some of the outliers.
- Train-Test Split (70:30).
- Imputed missing values and created new category ‘others’ and mode for categorical, median for numerical.
- Encoded variables: binary with 0,1; dummy for more than 2 classes.
- Applied MinMax Scaling on training data (excluding dummy variables).
- Created correlation heatmap

Feature Engineering and Model Building

- RFE selected top 15 features for the first Logistic Regression model.
- Total of 5 models are created out of which 5th model is the final model and based on P-values (accepted p-value < .05) and (VIF < 5) were checked simultaneously after each model.
- Model accuracy and Confusion Matrix were evaluated after the final model was build.

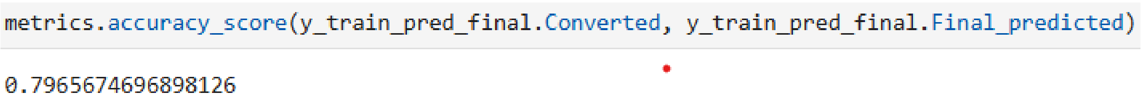
Model Evaluation

- Model 5 predicted probability on training data and using 0.5 as cutoff to determine target 0 or 1.



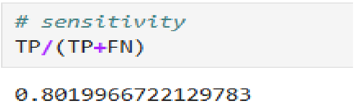
```
array([[3131,  816],
       [ 476, 1928]])
```

```
# Predicted     not converted    converted
# Actual
# not converted        3131      816
# converted            476      1928
```



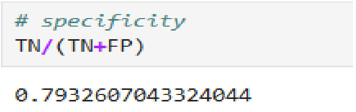
```
metrics.accuracy_score(y_train_pred_final.Converted, y_train_pred_final.Final_predicted)
```

```
0.7965674696898126
```



```
# sensitivity
TP/(TP+FN)
```

```
0.8019966722129783
```

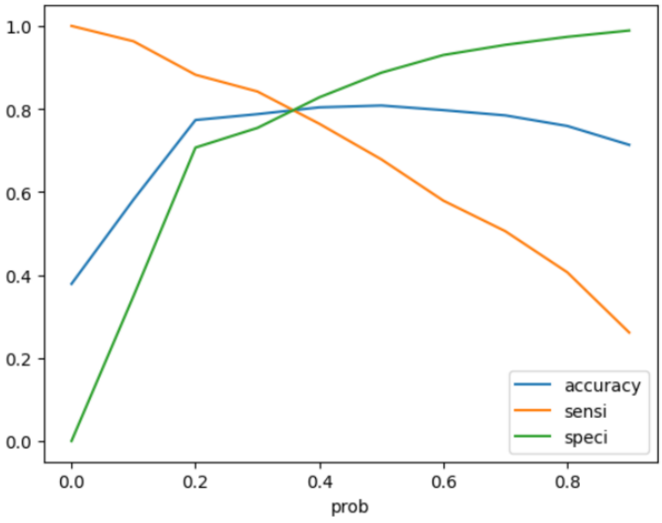


```
# specificity
TN/(TN+FP)
```

```
0.7932607043324044
```

Finding Optimal Probability Cutoff

- Specificity, sensitivity, and accuracy were calculated for different cutoffs, after determining we got optimal cutoff of 0.35.

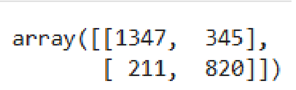


Prediction on Test Data

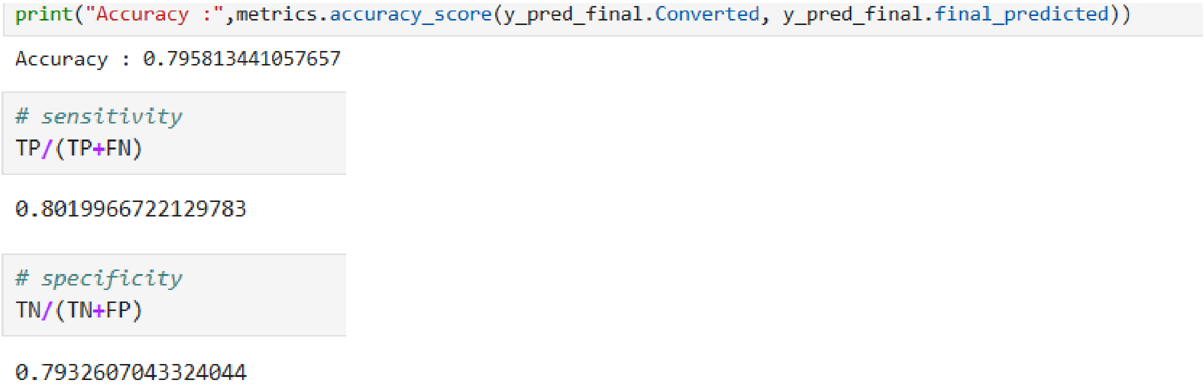
- We scaled the test data using MinMax Scaling. Model 5 predicted whether the target was 0 or 1 with a cutoff of 0.35. Then, we converted these probabilities into lead scores ranging from 0 to 100.

Model Evaluation on Test Data

- Everything same is done on test data:



```
array([[1347,  345],
       [ 211,  820]])
```



```
print("Accuracy :",metrics.accuracy_score(y_pred_final.Converted, y_pred_final.final_predicted))
```

```
Accuracy : 0.795813441057657
```

```
# sensitivity
TP/(TP+FN)
```

```
0.8019966722129783
```

```
# specificity
TN/(TN+FP)
```

```
0.7932607043324044
```

Learnings from Data Analysis

Data Cleaning

Handling missing values, removing unnecessary columns, and grouping rare categories improved data quality and model performance.

Feature Selection

Using RFE helped find important features, and VIF ensured no multicollinearity, making the model more efficient.

Outlier Detection

Removing outliers prevented skewed results and improved model accuracy.

Data Scaling

MinMax Scaling normalized the data, speeding up model training and preventing bias from features with larger ranges.

Optimal Cutoff

Setting the probability cutoff to 0.35 improved the model's specificity, sensitivity, and accuracy.

Model Iteration

Building and refining multiple models ensured the final one was accurate and well-validated.

Test Data Evaluation

Scaling test data and using the 0.35 cutoff led to accurate lead scoring.

These learnings show the importance of data preprocessing, feature selection, scaling, and iterative modeling in creating reliable predictive models.