



# LEAD SCORE CASE STUDY

## Powering X Education's Sales Transformation

case study

Submitted by:  
- Kanav Mehta



# X Education - Online Professional Training Platform

- Current Situation:
  - Generates leads through websites, search engines, and referrals
  - Existing lead conversion rate: Only 30%
  - Significant resource wastage on low-potential leads
  - Inefficient sales process

### Points to concentrate on:

- Time and effort spent on unproductive leads
- Missed opportunities with high-potential prospects
- Lack of systematic lead prioritization





# Our Strategic Solution - Lead Scoring Model

- **Objective: Develop a Predictive Lead Score**
- Model Highlights:
  - Scoring Range: 0-100
  - Purpose: Identify "Hot Leads" with high conversion potential
  - Goal: Improve conversion rate from 30% to 80%
- Key Deliverables:
  1. Logistic Regression Predictive Model
  2. Data-Driven Insights - Questionnaire
  3. Performance Visualization - PPT
  4. Actionable Recommendations - Summary
- Expected Outcomes:
  - Optimize sales team's efforts
  - Increase conversion efficiency
  - Reduce wasted resources
  - Systematic lead qualification process





# Methodology

- Importing Libraries & Setting up Analytics Environment
- Dataset Inspection
- Data Pre-Processing
- Exploratory Data Analysis
- Model Building – Logistic Regression
- Model Evaluation
- Predictions on Test Set
- Lead Score Generation
- Findings & Recommendations



# Dataset Inspection

- We start with 37 columns and over 9240 rows.
- Most of these columns are string, with only a handful of numerical features

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000	9240.000	9103.000	9240.000	9103.000	5022.000	5022.000
mean	617188.436	0.385	3.445	487.698	2.363	14.306	16.345
std	23405.996	0.487	4.855	548.021	2.161	1.387	1.811
min	579533.000	0.000	0.000	0.000	0.000	7.000	11.000
25%	596484.500	0.000	1.000	12.000	1.000	14.000	15.000
50%	615479.000	0.000	3.000	248.000	2.000	14.000	16.000
75%	637387.250	1.000	5.000	936.000	3.000	15.000	18.000
max	660737.000	1.000	251.000	2272.000	55.000	18.000	20.000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   Prospect ID                                                            9240 non-null   object
1   Lead Number                                                            9240 non-null   int64
2   Lead Origin                                                            9240 non-null   object
3   Lead Source                                                            9204 non-null   object
4   Do Not Email                                                           9240 non-null   object
5   Do Not Call                                                            9240 non-null   object
6   Converted                                                              9240 non-null   int64
7   TotalVisits                                                            9103 non-null   float64
8   Total Time Spent on Website                                           9240 non-null   int64
9   Page Views Per Visit                                                  9103 non-null   float64
10  Last Activity                                                          9137 non-null   object
11  Country                                                                6779 non-null   object
12  Specialization                                                         7802 non-null   object
13  How did you hear about X Education                                    7033 non-null   object
14  What is your current occupation                                       6550 non-null   object
15  What matters most to you in choosing a course                       6531 non-null   object
16  Search                                                                 9240 non-null   object
17  Magazine                                                              9240 non-null   object
18  Newspaper Article                                                     9240 non-null   object
19  X Education Forums                                                    9240 non-null   object
20  Newspaper                                                             9240 non-null   object
21  Digital Advertisement                                                 9240 non-null   object
22  Through Recommendations                                              9240 non-null   object
23  Receive More Updates About Our Courses                              9240 non-null   object
24  Tags                                                                  5887 non-null   object
25  Lead Quality                                                           4473 non-null   object
26  Update me on Supply Chain Content                                    9240 non-null   object
27  Get updates on DM Content                                             9240 non-null   object
28  Lead Profile                                                           6531 non-null   object
29  City                                                                  7820 non-null   object
30  Asymmetrique Activity Index                                           5022 non-null   object
31  Asymmetrique Profile Index                                           5022 non-null   object
32  Asymmetrique Activity Score                                           5022 non-null   float64
33  Asymmetrique Profile Score                                           5022 non-null   float64
34  I agree to pay the amount through cheque                             9240 non-null   object
35  A free copy of Mastering The Interview                               9240 non-null   object
36  Last Notable Activity                                                 9240 non-null   object
dtypes: float64(4), int64(3), object(30)
memory usage: 2.6+ MB
```



## DATA PREPROCESSING

- `Select` seems to be erroneously captured in the data collection process despite not being a valid data point.
- We replaced this with 'Unknown'

## NULLS

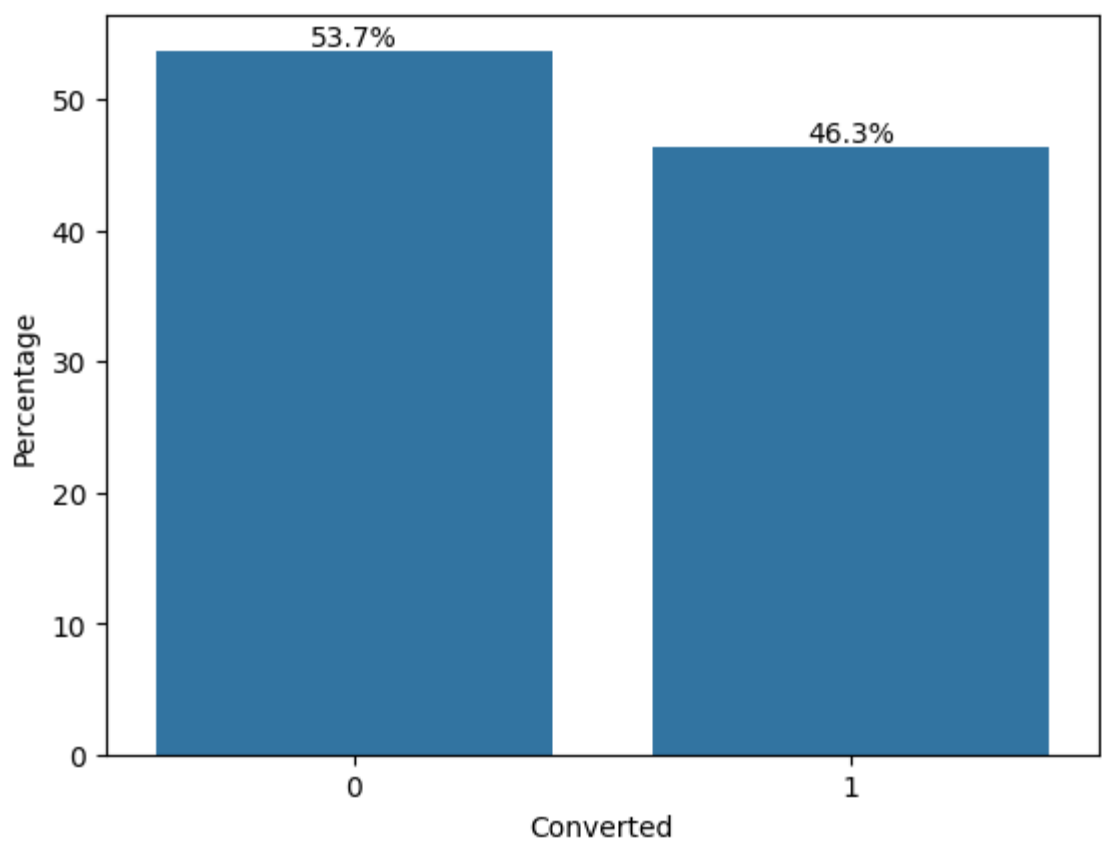
- We dropped features with Null % over 30%
- Retained 'TAGS' column despite high null% owing to its importance
- Dropped rows where 'TAGS' was Null.
- In low null columns
  - for Numerical Features – Imputed nulls with median
  - For Categorical Features – imputed nulls with mode
- Capped Outliers in Numerical features
- Reduced sub-categories in 'Lead Source'





# Exploratory Data Analysis

## Target Imbalance

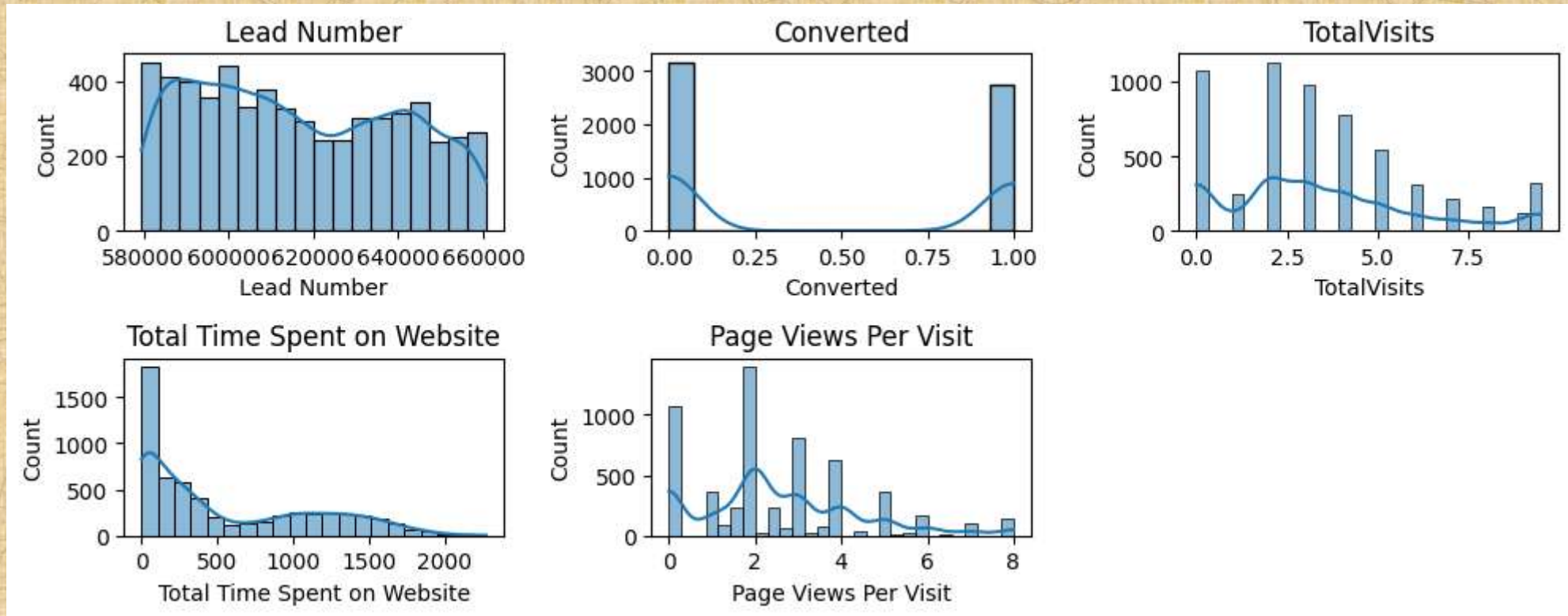


There is a slight imbalance in the Target variable in the given dataset.



# Exploratory Data Analysis

## Univariate Analysis - Numerical



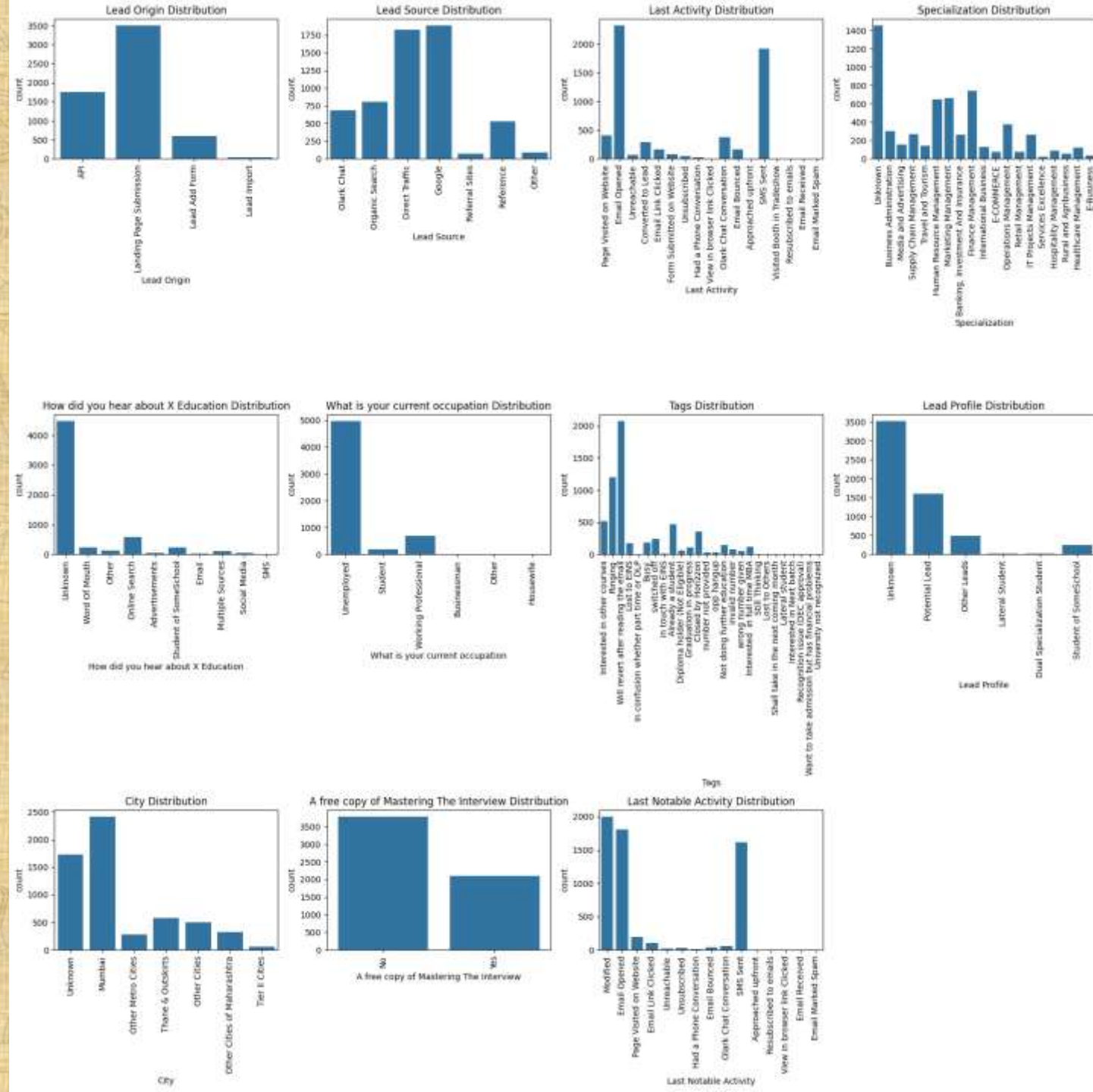
- We can see a slight skewness in the dataset
- It's a right tailed distribution for most of the numerical features.



# Exploratory Data Analysis

## Univariate Analysis - Categorical

- We can see a huge imbalance in most of the categorical features
- Some of these seem moderately balanced

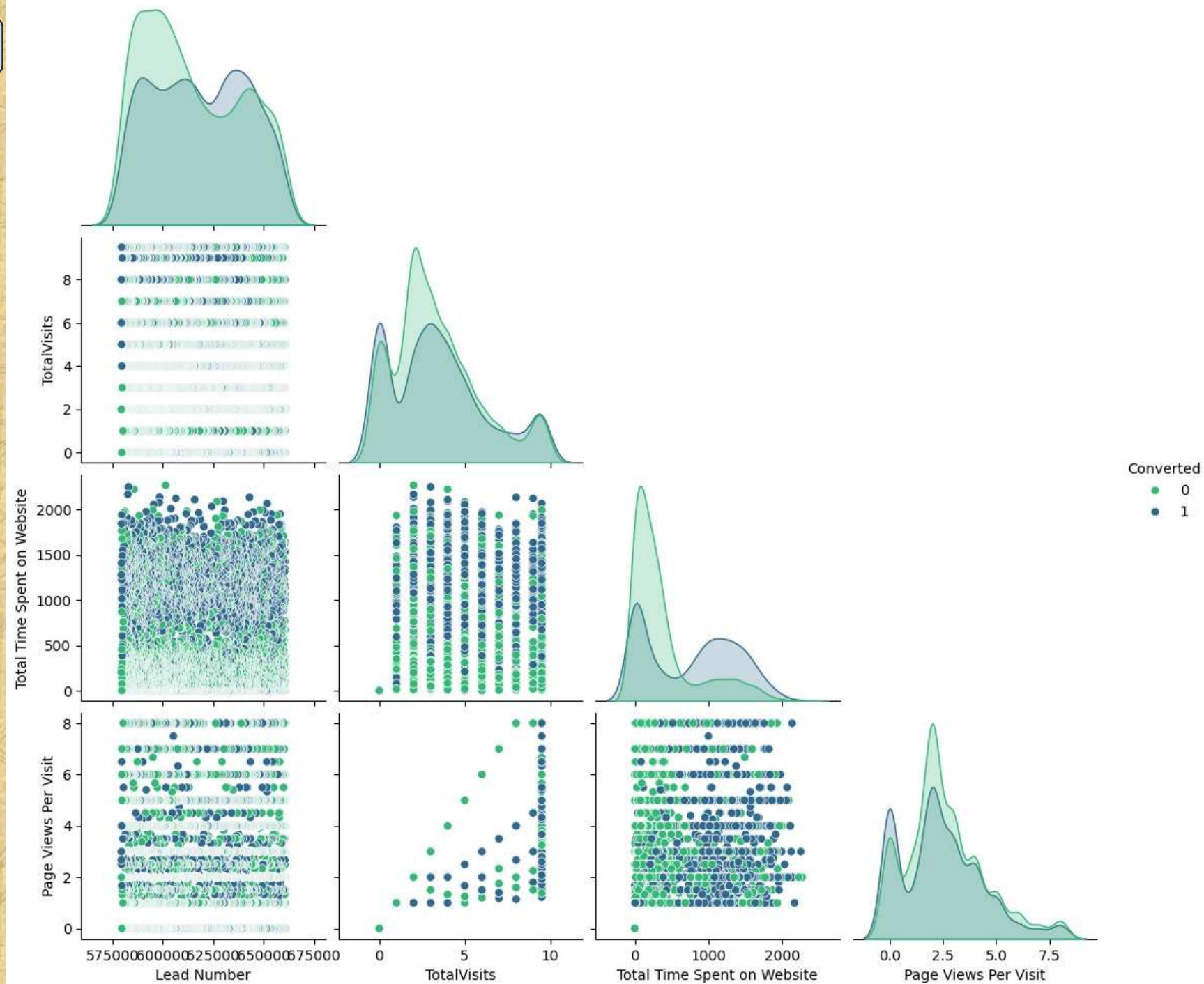




# Exploratory Data Analysis

## Bivariate Analysis - Numerical

- The only pair showing somewhat linear relationship is between -  
`TotalVisits` & `Page Views Per Visit`





# Exploratory Data Analysis

## Multivariate Analysis - Numerical

- A high correlation can be seen between `Page Views Per Visit` & `Total Time Spent on Website`
- A good Correlation can also be seen between `Total Time Spent on Website` & `Converted`
- This could imply that those who are highly interested to buy an education program visit the website often, or spend more time exploring the programs during their visits.





# Model Building – Logistic Regression

- We start with one-hot encoding the categorical columns
- We get 112 columns as a result
- Here we have a corr heatmap of all dummy features

```
# Converting categorical variables into dummy variables (one-hot encoding)  
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True, dtype=int)
```

✓ 0.0s

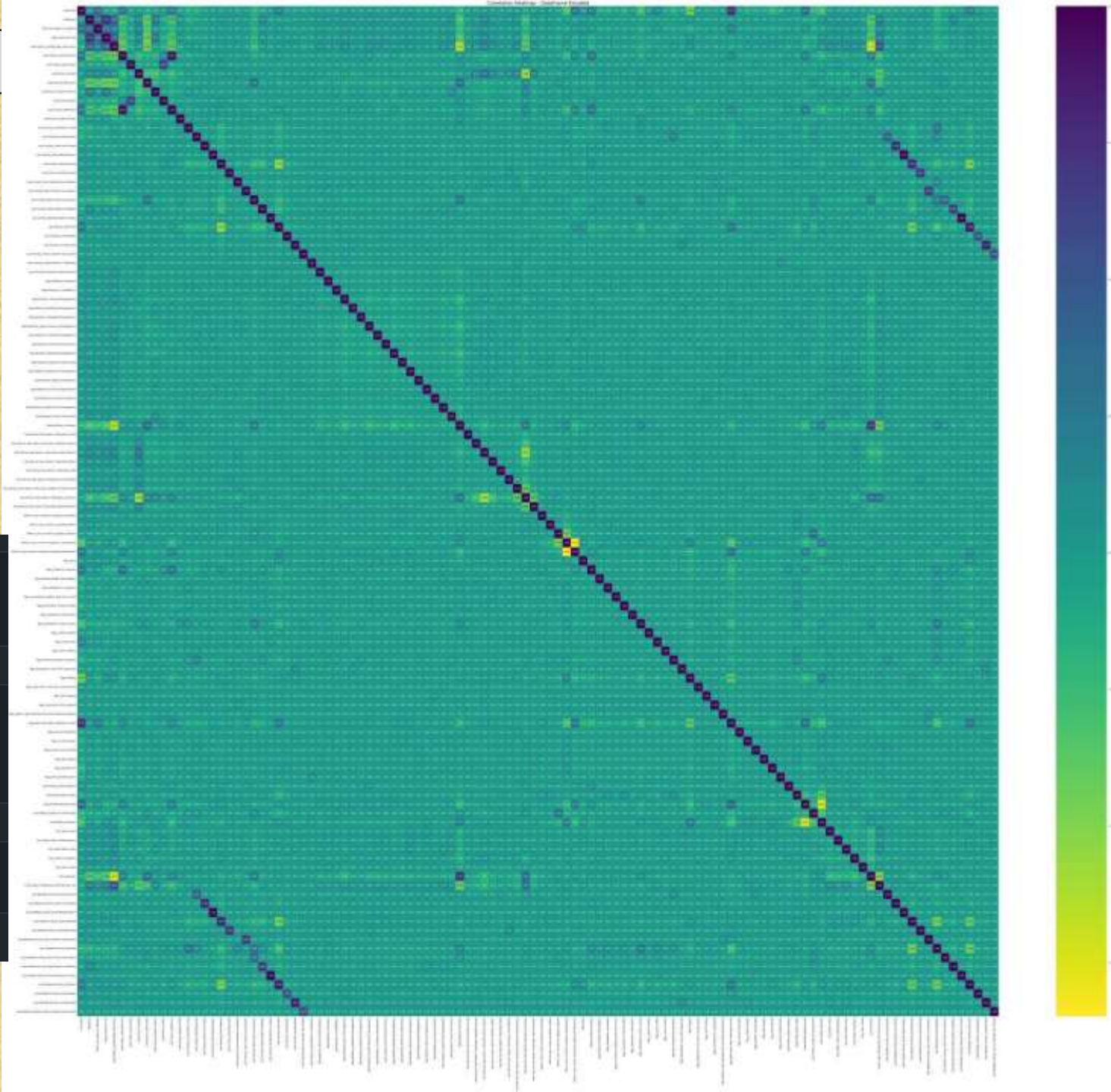
```
# Setting 'Lead Number' as the DataFrame index  
df = df.set_index('Lead Number', drop=True)  
numerical_cols.remove('Lead Number')
```

✓ 0.0s

```
df.shape
```

✓ 0.0s

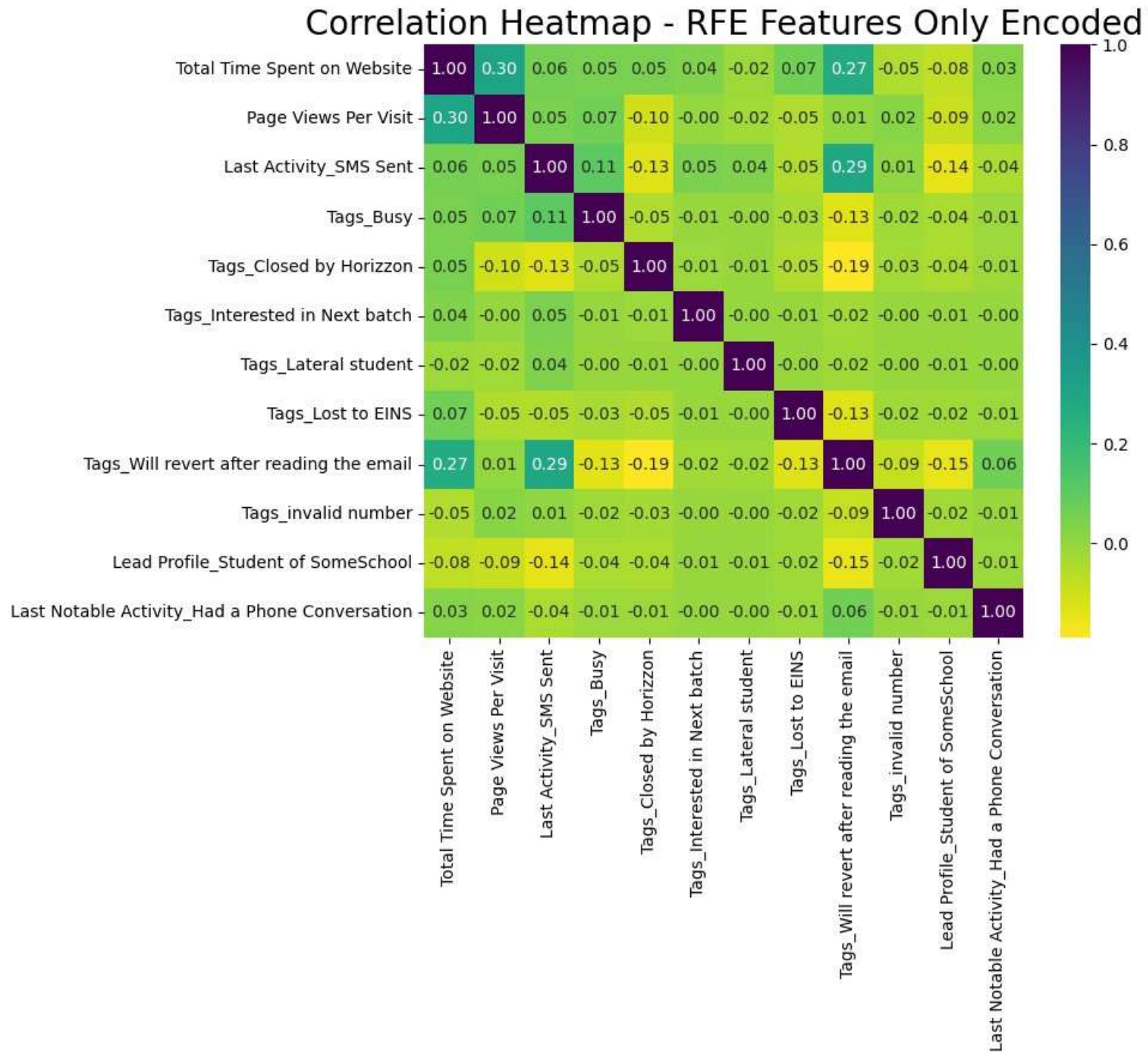
```
(5887, 112)
```





## Train-Test Split, Scaling & RFE

- We split the data into train & test sets
- Scale the Numerical features using MinMaxScaler
- Using RFE to quickly filter down 12 features for analysis
- We don't see extremely high correlation between features here, but we'll manually check using statsmodels





## Final Model

- At the end of the 5<sup>th</sup> model, we have no longer any feature with high p-values or VIFs
- We stop dropping any more features and are left with 8 features

	feature	VIF
0	const	4.420
7	Tags_Will revert after reading the email	1.353
1	Total Time Spent on Website	1.261
2	Page Views Per Visit	1.158
3	Last Activity_SMS Sent	1.140
5	Tags_Closed by Horizzon	1.106
4	Tags_Busy	1.066
8	Lead Profile_Student of SomeSchool	1.056
6	Tags_Lost to EINS	1.055

### Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	4709
Model:	GLM	Df Residuals:	4700
Model Family:	Binomial	Df Model:	8
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-580.59
Date:	Tue, 17 Dec 2024	Deviance:	1161.2
Time:	22:03:35	Pearson chi2:	4.05e+03
No. Iterations:	8	Pseudo R-squ. (CS):	0.6788
Covariance Type:	nonrobust		
	coef	std err	z P> z  [0.025 0.975]
const	-4.4580	0.215	-20.777 0.000 -4.879 -4.037
Total Time Spent on Website	3.4602	0.347	9.966 0.000 2.780 4.141
Page Views Per Visit	-1.1929	0.375	-3.177 0.001 -1.929 -0.457
Last Activity_SMS Sent	1.4433	0.179	8.076 0.000 1.093 1.794
Tags_Busy	3.3894	0.229	14.799 0.000 2.940 3.838
Tags_Closed by Horizzon	9.5875	1.017	9.423 0.000 7.593 11.582
Tags_Lost to EINS	7.7425	0.634	12.214 0.000 6.500 8.985
Tags_Will revert after reading the email	6.9136	0.207	33.392 0.000 6.508 7.319
Lead Profile_Student of SomeSchool	-2.3014	0.907	-2.537 0.011 -4.080 -0.523



## Model Evaluation – Metrics – Train Set

```
Training Performance:
              precision    recall  f1-score   support

     0       0.96      0.97      0.96      2502
     1       0.96      0.96      0.96      2207

   accuracy          0.96      0.96      0.96      4709
  macro avg          0.96      0.96      0.96      4709
weighted avg          0.96      0.96      0.96      4709

Confusion Matrix (Training):
[[2416   86]
 [  90 2117]]
```

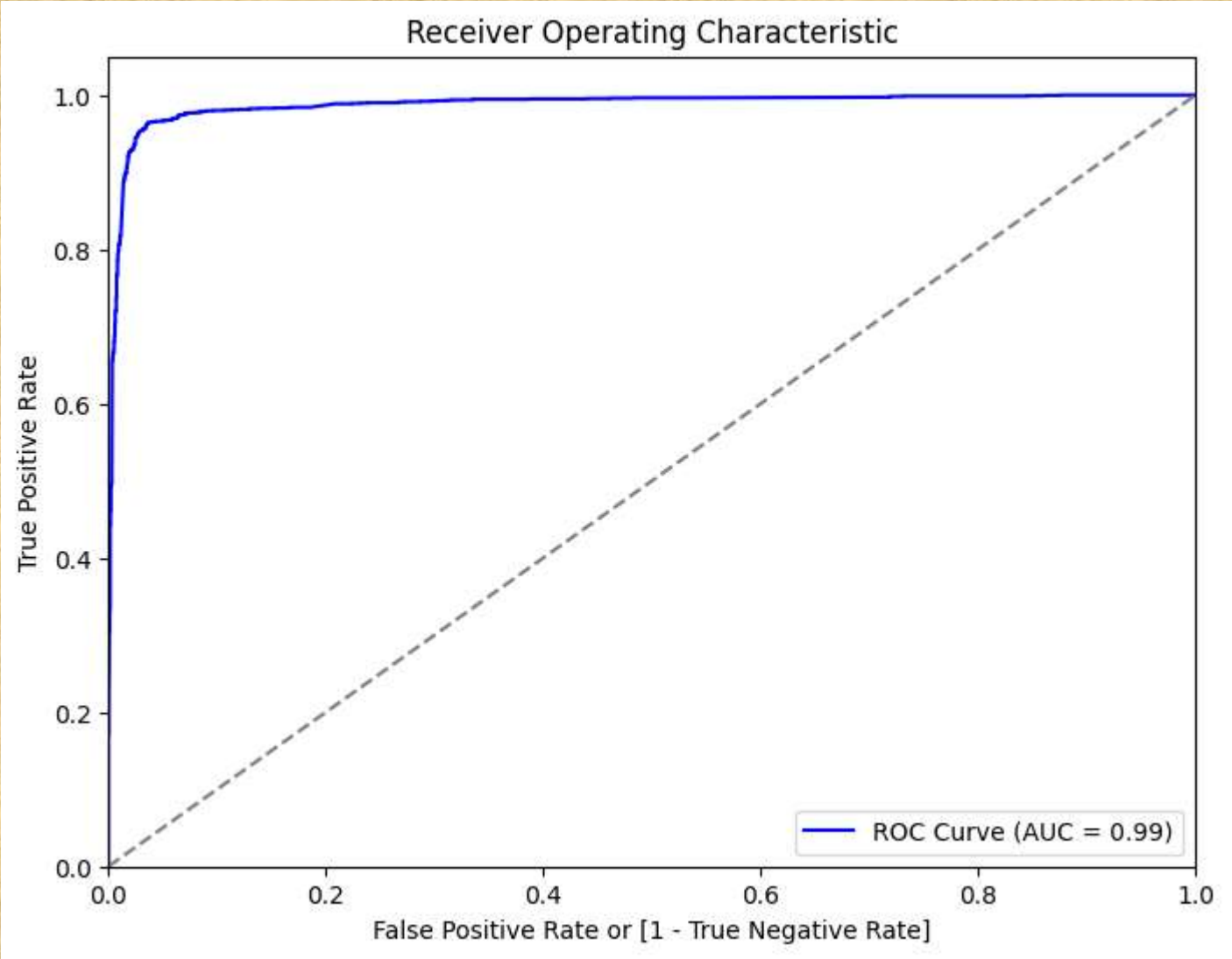
Accuracy	0.9626
Sensitivity (Recall)	0.9592
Specificity	0.9656

- We take a look at the Classification Report & Confusion Matrix of the Train Set
- Cross-Validation Scores: [0.96178344 0.96496815 0.95329087 0.96815287 0.95855473]
- Mean CV Accuracy: 96.14% (+/- 1.03%)



# Model Evaluation – ROC AUC – Train Set

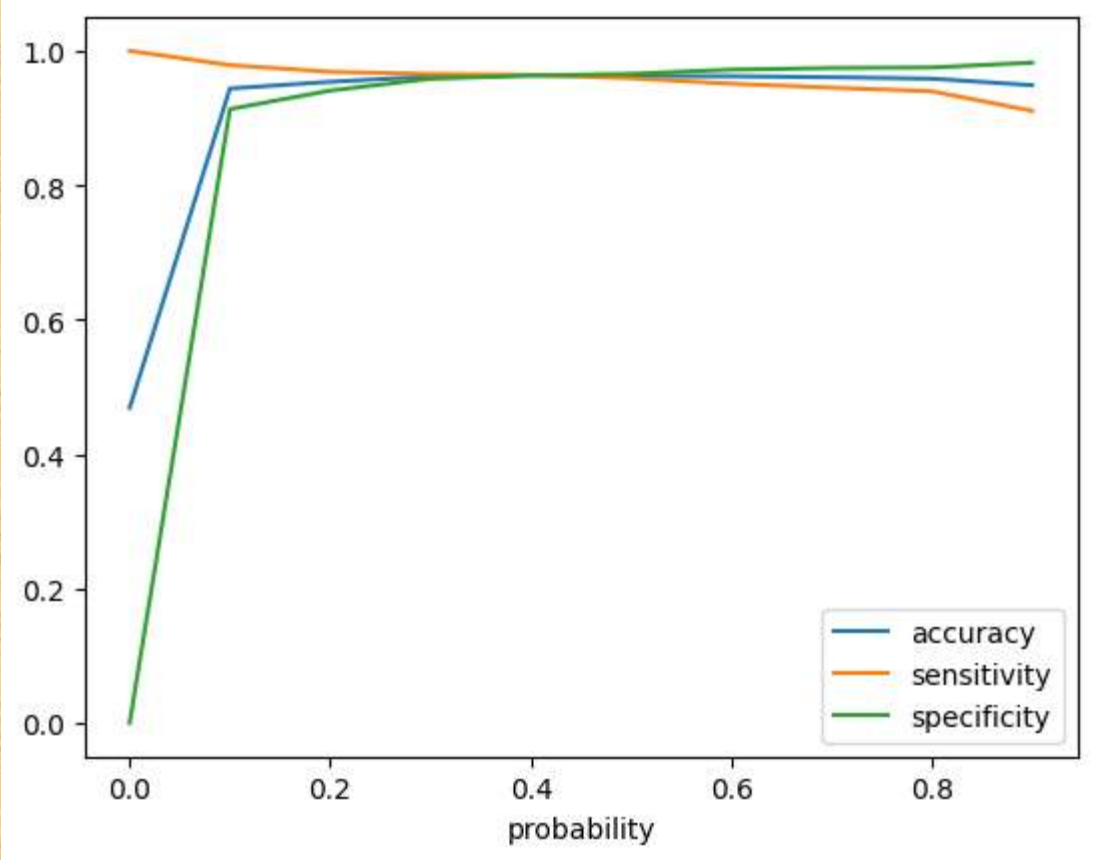
- The ROC curve with an AUC of 0.99 indicates that the logistic regression model is performing exceptionally well.
- This means the model is highly accurate in distinguishing between positive and negative classes. It has a strong ability to correctly classify instances into their respective categories.





# Optimal Cutoff – Accuracy-Sensitivity-Specificity

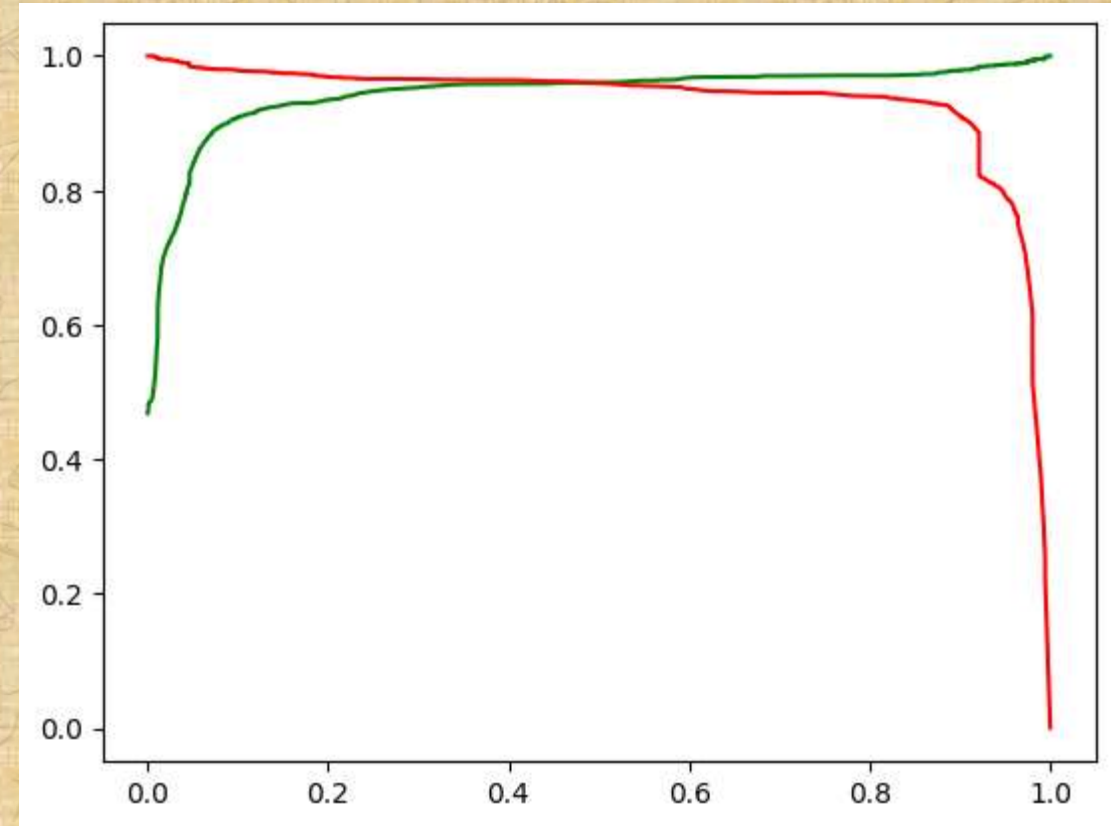
- We can see that all 3 curves intersect at about 0.4
- The accuracy at this threshold is 0.9637





## Optimal Cutoff – Precision-Recall

- We can see that all Precision & Recall intersect at about 0.45
- The accuracy at this threshold is 0.9635





# Predictions on Test Set – Evaluation Metrics

- We check for Accuracy on Test Set using both thresholds we found in the earlier sliders
- The ‘Accuracy-Sensitivity-Specificity’ threshold of 0.4 gives slightly higher accuracy in Test set, so we’ll proceed with this value.

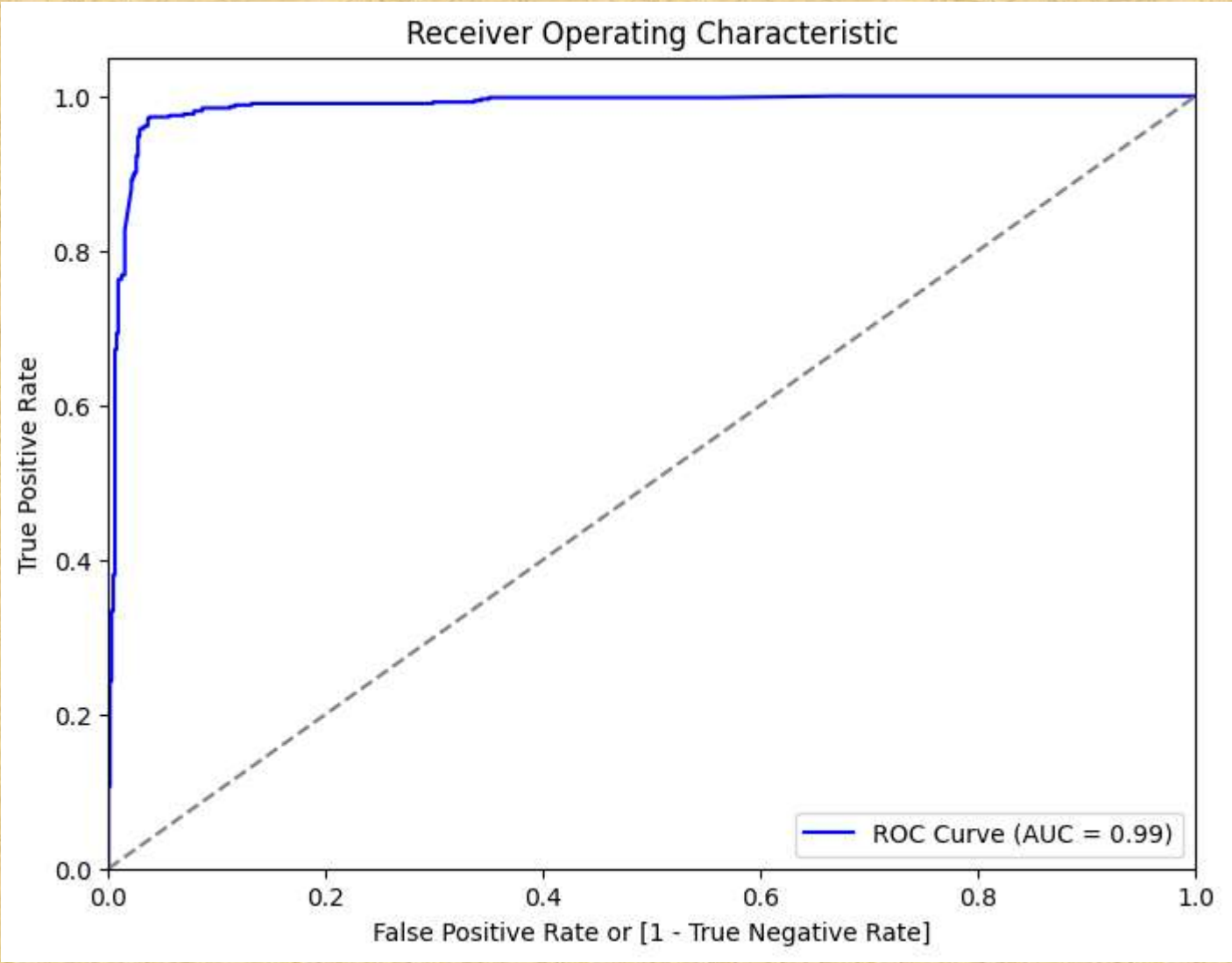
Testing Performance:					
	precision	recall	f1-score	support	
0	0.98	0.96	0.97	660	
1	0.95	0.97	0.96	518	
accuracy			0.97	1178	
macro avg	0.97	0.97	0.97	1178	
weighted avg	0.97	0.97	0.97	1178	
Confusion Matrix (Testing):					
[[635 25]					
[ 14 504]]					

Accuracy	0.9669
Sensitivity (Recall)	0.973
Specificity	0.9621



# Model Evaluation – ROC AUC – Test Set

- In the Test Set we see an ROC curve with an AUC of 0.99.
- This means the model is highly accurate in distinguishing between positive and negative classes and can correctly classify instances into their respective categories.





## Lead Score & Priority Labels

- Finally, we assign Lead Scores to each Lead
- Lead Score is basically the probability of the Lead to Convert multiplied by 100
- We also categorized the Leads as – Very High, High, Medium & Low Priority – based on their Lead Scores
- priority level based on a lead score:
  - Score > 80: Very High
  - Score > 60: High
  - Score > 40: Medium
  - Score  $\leq$  40: Low
- Higher scores indicate higher priority levels.





## Key Findings

- **Overall Accuracy:** 96.14% (Mean CV Accuracy) on the training set, with consistent performance on the test set
- **ROC AUC Score:** 0.99, indicating excellent discrimination between converted and non-converted leads
- **High Sensitivity (Recall):** 95.92%, demonstrating strong ability to identify actual conversions
- **High Specificity:** 96.56%, showing robust performance in correctly identifying non-converting leads
- **Optimal Probability Threshold:** Identified at 0.4 using Accuracy-Sensitivity-Specificity curve analysis
- **Feature Significance:** Successfully reduced feature set while maintaining high predictive performance



## Recommendations

- 1. Predictive Insights:** Use the model's output to assign lead scores, enabling the sales and marketing teams to prioritize high-probability leads effectively.
- 2. Periodic Model Validation:** Continuously retrain the model with updated data to ensure its performance remains aligned with evolving customer behaviors and market trends.
- 3. Optimize Campaign Strategies:** Focus marketing and engagement efforts on activities or segments associated with high conversion probabilities as identified by the model.
- 4. Monitor Key Metrics:** Conduct regular evaluations of the model's sensitivity, specificity, and accuracy to ensure consistent performance.
- 5. Iterate and Enhance:** Explore additional features, such as external data sources or behavioral metrics, to further refine the model's predictive capabilities.
- 6. Strategic Use of Thresholds:** Adjust the probability threshold based on specific business goals, such as increasing conversion rates or minimizing false negatives, to optimize resource allocation.