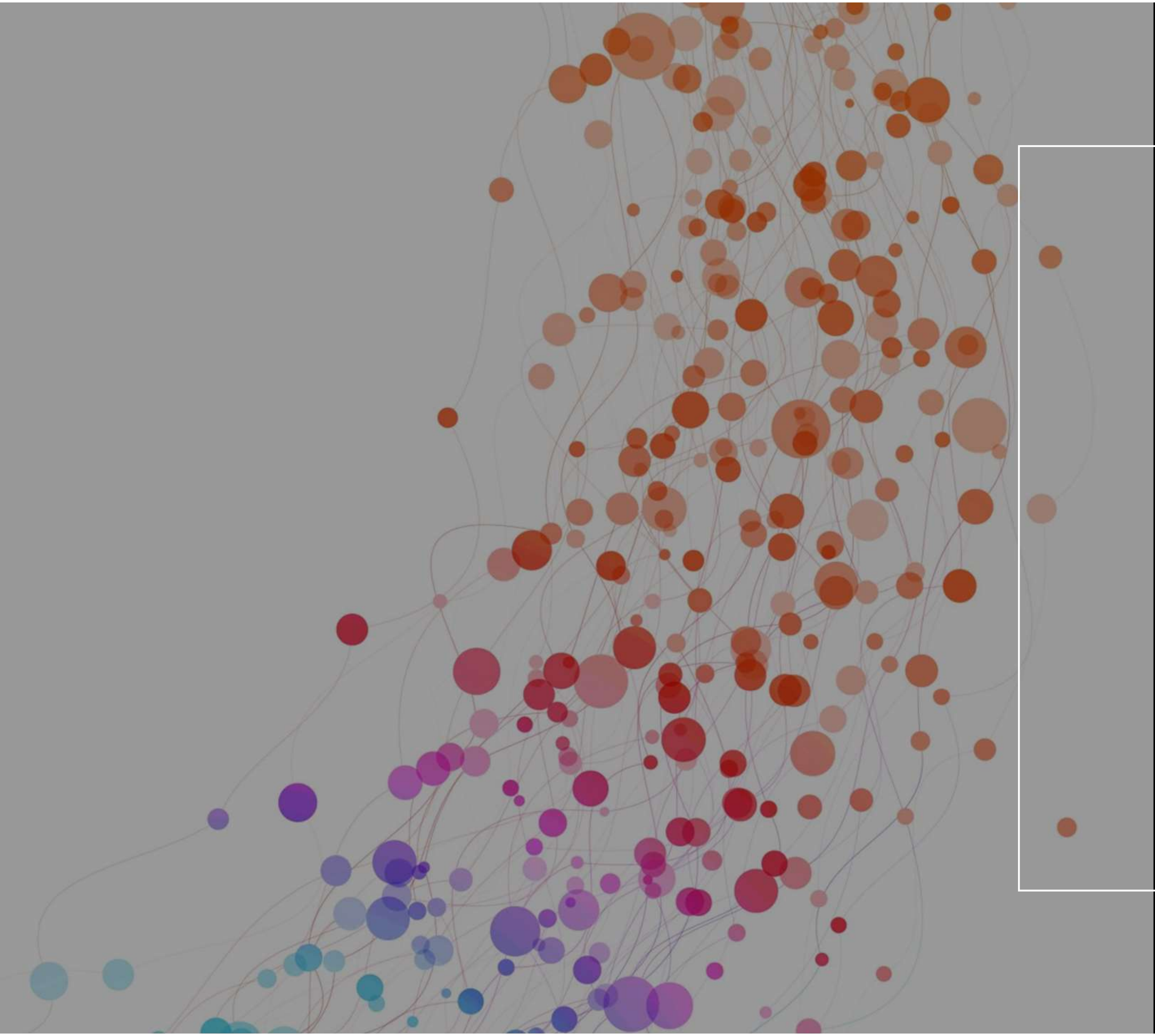


LEAD SCORING CASE STUDY

VIJAY ATMURI
AVISEK HALDER
AYUSH KURIA

PROBLEM SOLVING APPROACH

- Business Understanding
- Problem Mapping and Solution Approach
- Data Understanding
- EDA
- Data Preparation
- Model Building
- Model Evaluation
- Model Prediction

BUSINESS UNDERSTANDING

An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.

The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

Now, although X Education gets a lot of leads, its lead conversion rate is very poor. For example, if, say, they acquire 100 leads in a day, only about 30 of them are converted. To make this process more efficient, the company wishes to identify the most potential leads, also known as 'Hot Leads'. If they successfully identify this set of leads, the lead conversion rate should go up as the sales team will now be focusing more on communicating with the potential leads rather than making calls to everyone.

PROBLEM MAPPING & SOLUTION APPROACH

The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with a higher lead score have a higher conversion chance and the customers with a lower lead score have a lower conversion chance. The CEO has given a ballpark of the target lead conversion rate to be around 80%.

- The problem can be mapped to find the hot leads i.e. the leads that are most likely to convert into paying customer.
- This is a classification problem, where in we need to build a model to assign lead score to each of the leads such that the customer with higher lead score has higher conversion chance and customer with lower lead score has lower conversion chance
- The company wants to utilize their resources optimally such that the lead conversion rate to be around 80% (we need to have suitable cutoff of the lead score and identify the potential leads such that lead conversion rate (recall_score) would be greater than 80%)

DATA UNDERSTANDING

- Imported the data from 'Leads.csv' file using pandas, pd.read_csv() with the name as 'data'.
- The 'Converted' column is the target variable, i.e., hot leads
- There are total of 37 columns and 9240 rows
- From the info , we see that there are columns with missing values and columns with values as 'Select'. These 'Select Values' has been treated as missing values.
- There is missing data for Lead Quantity and Lead Profile which are assigned by the employees based the lead profile and intuition
- The data set is not so biased as we see good sample containing 38% of lead conversion

Conversion ratio

```
data['Converted'].value_counts()[1] / data.shape[0]
```

```
0.3853896103896104
```

EDA

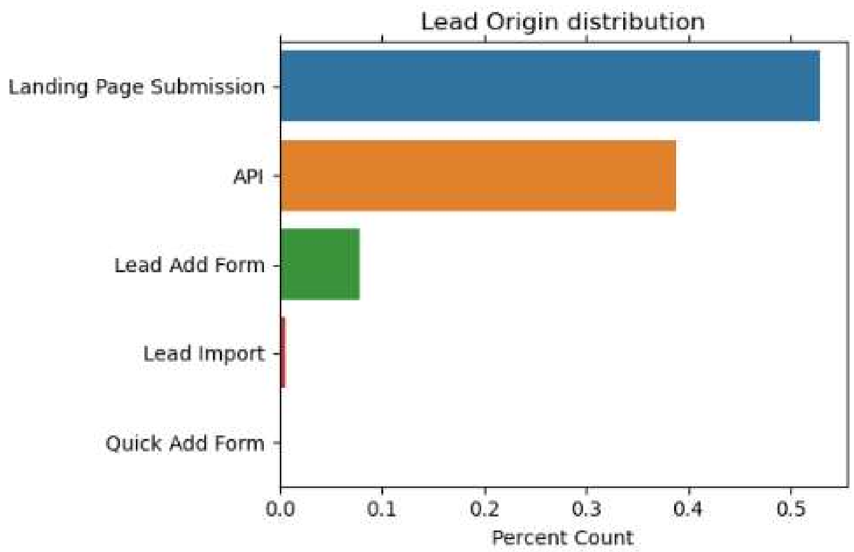
- Missing Values Treatment
  - Removed columns with more than 45% of the values are null (np.NaN)
  - The Last Notable Activity values are same as Last Activity, expect for 'Modified'. But since Last Activity has missing values, dropped this column
  - We cannot impute the mode / mean for the 29% of the missing data in Country and City. Further there are data inconsistencies between Country and City. Hence dropped.
  - The missing values for columns 'Tags', 'What is your current occupation', 'Specialization' has been imputed as 'Missing' as imputing with mean / mode seems to add bias
  - The missing values in numerical variables are imputed with Median, as we see outliers in these columns

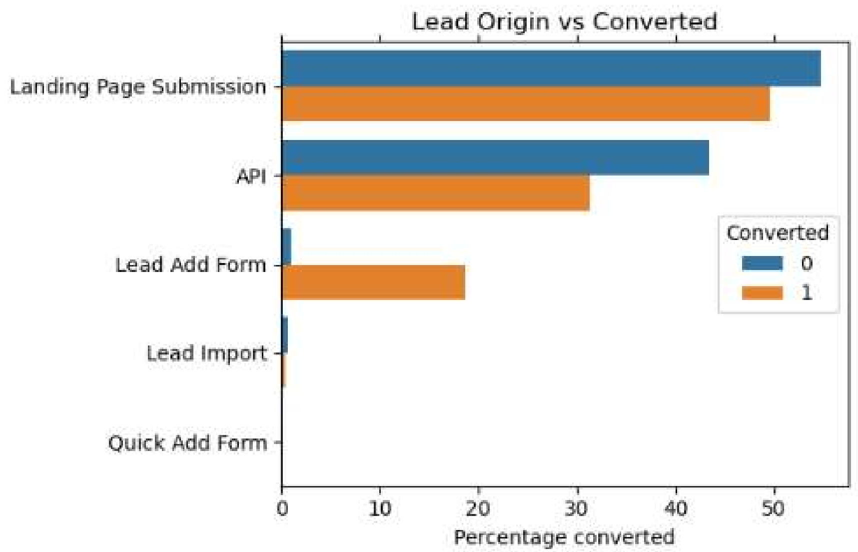
Outlier Treatment

- Outliers are detected in 'TotalVisits' and 'Page Views Per Visit'. Removed the rows that are above identified threshold values as outlier treatment

UNIVARIATE ANALYSIS

Univariate Analysis for Categorical Columns & Numerical columns are performed with respective to their overall distribution and their contribution towards lead conversion. Refer the .ipynb file for more details



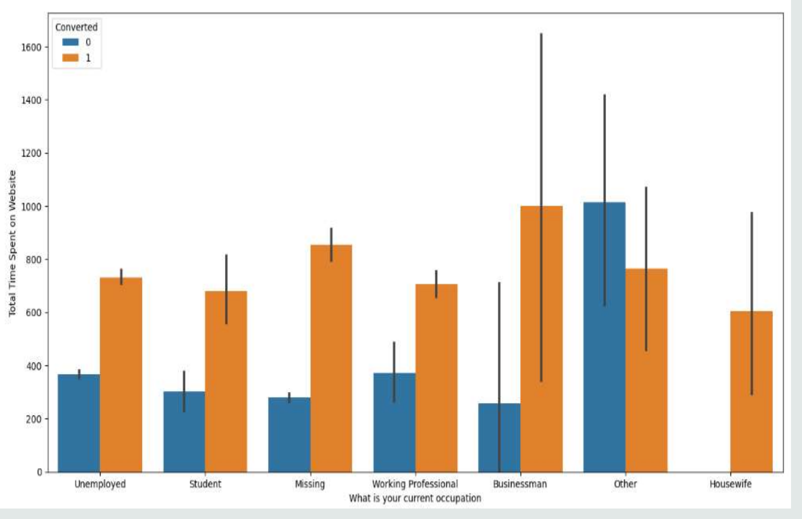


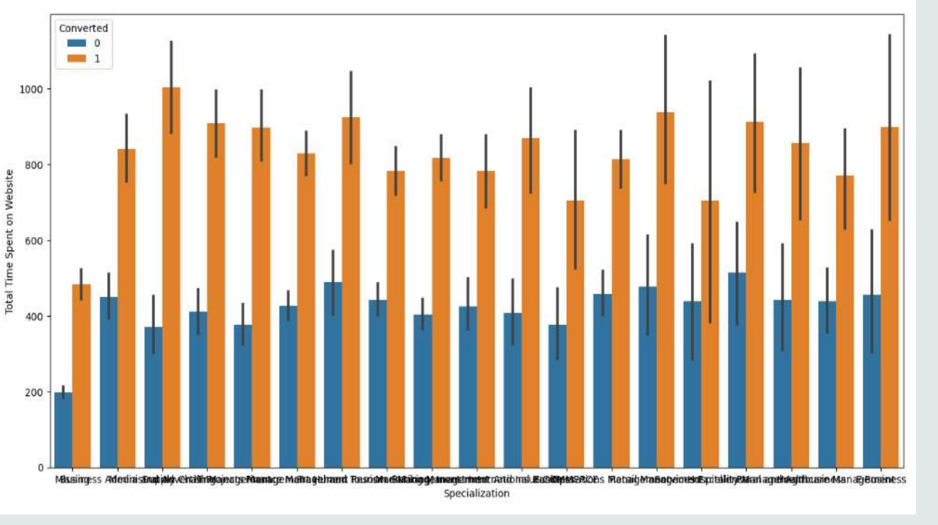
Inference from Lead Origin

1. 'Landing Page Submission' contributes highest pertage of leads followed by 'API'. 'Lead Import' and 'Quick Add Form' has negligible contribution
2. Even though Lead Add Form has negligible overall contribution (10%), it significantly contributes to the Lead Conversion (20%)
3. 'Landing Page Submission' has almost similar contirbution of non-conversion and conversion of leads
4. 'API' had high non-converted leads compared to converted leads

BI-VARIATE ANALYSIS

Bi-Variate Analysis was performed between different columns as below





Inference:

As the amount of time spent on website increased the lead has high chances of conversion. The business can focus on leads spending more time on the website

HEAT MAP

Average Time Spent on Website has a positive correlation with Converted Leads

Page Views per Visit and Average Time spend on Website has negative correlation

DATA PREPARATION

- Converted binary variables (Yes/No) to 0/1, for the columns 'Do Not Email', 'Through Recommendations', 'A free copy of Mastering The Interview'
- For categorical variables with multiple levels, created dummy features (one-hot encoded)
- Created a new variable - Average time spent as (Total Time Spent on Website / TotalVisits) and dropped these two columns
- The dataset was split into train (70%) and test(30%) using sklearn.model_selection.train_test_split
- Some of the columns have a different value ranges compared to other binary, dummy variables. Performed Feature Scaling using MinMaxScaler() for the columns 'Average Time Spent on Website','Page Views Per Visit' and performed scalar.fit_transform() on training data set.
- Dropped highly correlation dummy variables as observed for correlation matrix

MODEL BUILDING

- We have used statsmodels.api and GLM model for binomial families to build the model, provides more detailed statistical output, including parameter estimates, standard errors, p-values, confidence intervals, and model fit statistics like AIC and BIC. This can be useful for statistical inference and interpretation.
- After creating dummy variables, we have around 97 features that are fed into the model. This may lead to over fitting and multi-collinearity. And also, difficult to interpret the important features.
- We have then proceeded with Feature Selection using Recursive Feature Elimination (RFE) and selected 25 features for further model building
- We continued with the further feature elimination using the top to bottom approach, where in we eliminated the feature based on higher p-values (lower significance) and high VIF value (high multi-collinearity)

MODEL EVALUATION

The final model has 10 features and below are the statistics

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6458
Model:	GLM	Df Residuals:	6447
Model Family:	Binomial	Df Model:	10
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1730.6
Date:	Mon, 18 Mar 2024	Deviance:	3461.1
Time:	01:03:14	Pearson chi2:	7.00e+03
No. Iterations:	8	Pseudo R-squ. (CS):	0.5519
Covariance Type:	nonrobust		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.1443	0.060	-19.013	0.000	-1.262	-1.026
Do Not Email	-1.2043	0.208	-5.783	0.000	-1.612	-0.796
Average Time Spent on Website	4.9904	0.386	12.937	0.000	4.234	5.746
Lead Source_Welingak Website	6.0742	1.028	5.911	0.000	4.060	8.088
Tags_Closed by Horizzon	6.9748	0.718	9.710	0.000	5.567	8.383
Tags_Interested in other courses	-1.7310	0.323	-5.360	0.000	-2.364	-1.098
Tags_Lost to EINS	5.4357	0.523	10.403	0.000	4.412	6.460
Tags_Ringing	-2.3631	0.201	-11.755	0.000	-2.757	-1.969
Tags_Will revert after reading the email	4.8365	0.170	28.416	0.000	4.503	5.170
Tags_switched off	-2.8750	0.588	-4.893	0.000	-4.027	-1.723
Last Notable Activity_Modified	-1.7847	0.112	-15.993	0.000	-2.003	-1.566

	Features	VIF
1	Average Time Spent on Website	1.41
9	Last Notable Activity_Modified	1.38
7	Tags_Will revert after reading the email	1.21
4	Tags_Interested in other courses	1.10
0	Do Not Email	1.09
3	Tags_Closed by Horizzon	1.08
6	Tags_Ringing	1.05
5	Tags_Lost to EINS	1.04
2	Lead Source_Welingak Website	1.01
8	Tags_switched off	1.01

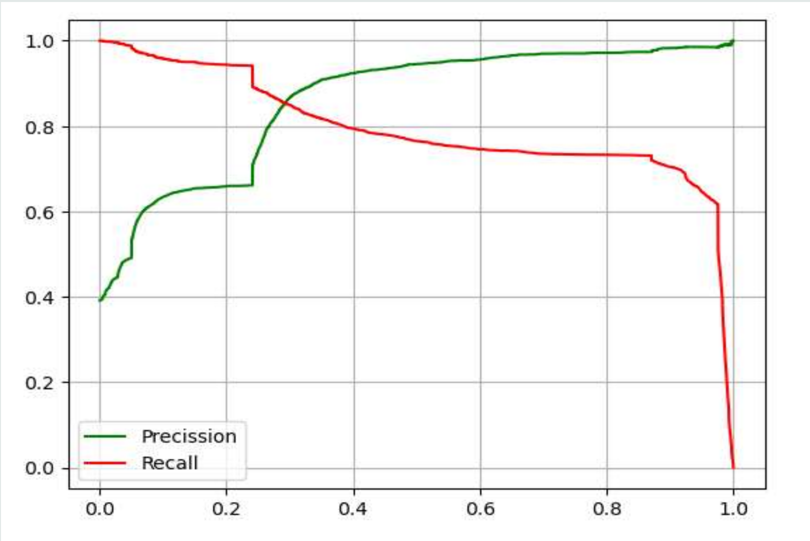
EVALUATION METRICS

We have considered the ACCURACY, RECALL and PRECISION as evaluation metrics.

Since the CEO expectation was to have 80% lead conversion, our target metric is RECALL to be at least 80%.

Recall - The ratio of true positive to Actual positive (We need to have high sensitivity such we do not miss on any lead that can be converted)

The optimum cut-off value is found to be 0.3 (i.e. Lead Score of 30%) from the Precision_Recall_Curve

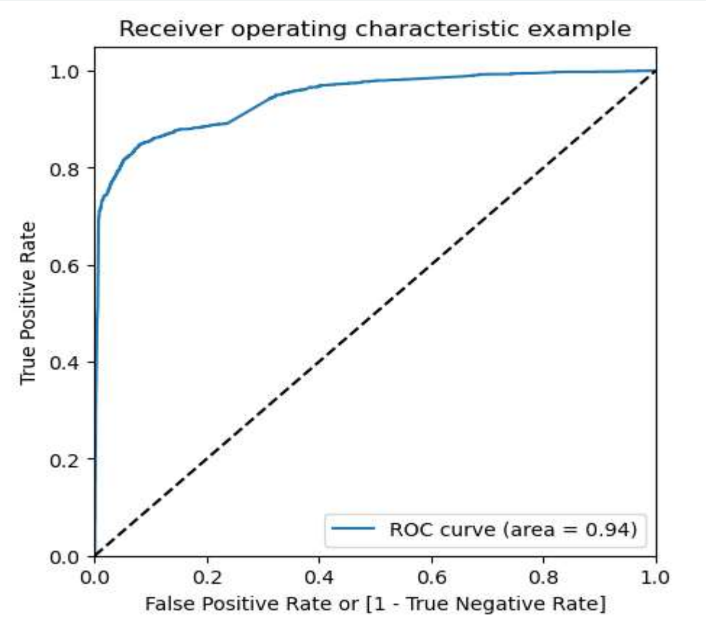


EVALUATION METRICS

Plotting ROC Curve

An ROC curve demonstrates several things:

- It shows the trade-off between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.



The ROC Curve area is 0.94, indicating higher accuracy of the model

EVALUATION METRICS

Below are the model evaluation metrics on the training data set

Accuracy – 89%

Recall – 85%

Precision – 89%

F1-Score – 86%

Classification Report of the Logistic Regression Model

	precision	recall	f1-score	support
0	0.90	0.92	0.91	3930
1	0.87	0.85	0.86	2528
accuracy			0.89	6458
macro avg	0.89	0.88	0.88	6458
weighted avg	0.89	0.89	0.89	6458

EVALUATION METRICS

Cross Validation Score

Evaluated the model performance using cross validation technique, to see the model metrics on unseen data, with cross validation folds as 10.

	count	mean	std	min	25%	50%	75%	max
fit_time	10.0	0.024519	0.003554	0.020928	0.021927	0.023424	0.026906	0.031894
score_time	10.0	0.007128	0.000661	0.005977	0.006976	0.006979	0.007719	0.007973
test_accuracy	10.0	0.889131	0.010468	0.871517	0.886180	0.889319	0.894616	0.904025
test_precision	10.0	0.950976	0.019493	0.913462	0.954486	0.958061	0.960048	0.970297
test_recall	10.0	0.755934	0.016355	0.739130	0.743820	0.750988	0.763140	0.790514
test_f1	10.0	0.842208	0.014830	0.818381	0.836488	0.840929	0.849890	0.865801

The above metrics are obtained by sklearn.cross_validation considering a default threshold of 0.5.

And looking at the results we see that values are closely bound indicating the model performance is in a close range and near to the model evaluation metrics and performs descent on an unseen data

MODEL PREDICTION

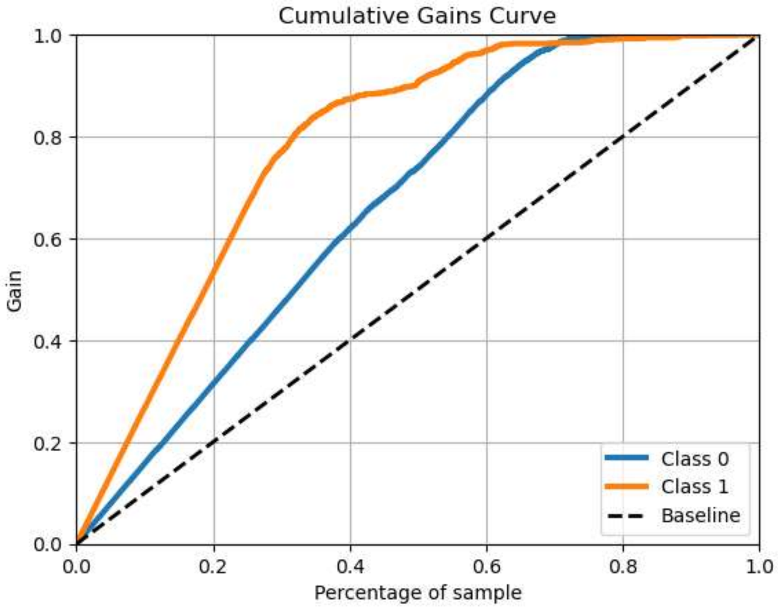
- Applied the scalar.transform() on the test data set to transform the test data using feature scaling metric fitted and transformed on the train data set
- Below are the model metrics on the test data set and are comparable to the model performance on train data set, indicating that there is no overfitting by the model

Classification Report of the Logistic Regression Model

	precision	recall	f1-score	support
0	0.92	0.91	0.91	1737
1	0.85	0.87	0.86	1031
accuracy			0.89	2768
macro avg	0.88	0.89	0.88	2768
weighted avg	0.89	0.89	0.89	2768

MODEL PREDICTION

Cumulative Gains Curve



- The Class 1 curve i.e. the positive class probability (Lead Conversion) is far away from the base line
- From the Gain Curve, we can see that by contacting top 30% of the customers (sorted list of predicted probabilities) would result is approaching 80% of the Leads who are likely to convert

KS – Statistic

```
print("KS Statistic:", ks_statistic)
```

```
KS Statistic: 0.3782514450867052
```

- A good model will have KS Statistic 40% or more. And current model is near to the good value

MODEL PREDICTION

Lead Score for test data set

```
# Assigning the lead score to each of the leads based on model prediction
Lead_Score = round((res.predict(sm.add_constant(X_test)))*100,2)
```

```
Lead_Score.head()
```

```
3224     3.62
4864     0.50
4937     3.49
7987    96.35
1641    24.15
dtype: float64
```

Final Features of the Model

	coef
const	-1.1443
Do Not Email	-1.2043
Average Time Spent on Website	4.9904
Lead Source_Welingak Website	6.0742
Tags_Closed by Horizzon	6.9748
Tags_Interested in other courses	-1.7310
Tags_Lost to EINS	5.4357
Tags_Ringing	-2.3631
Tags_Will revert after reading the email	4.8365
Tags_switched off	-2.8750
Last Notable Activity_Modified	-1.7847

RECOMMENDATIONS

1. The lead score predicted by the model indicates the likelihood of the lead to convert. A high score indicates Hot-Leads and low score indicates Cold-Leads
2. Using the model, the business can identify the hot leads that have high chance of conversion and thereby efforts of the Sales team can be channelized to concentrate on high probable hot leads rather than following up with each lead
3. The model accuracy depends on the Tags assigned to the leads. So it is important that tags are assigned accurately
4. The business team can accordingly set a base level for the Lead Score depending on their targets and reach out to the prospective leads
5. From the model the top 3 features contributing significantly towards lead conversion are
    a. Tags
    b. Lead Source
    c. Average Time Spend on Website

6. The model accuracy can be further improved during the novice techniques like Gradient Boosting, Simple Vector Machine, NNN