Lead Scoring Case Study
SUBMISSION REPORT

- *To build a Logistic Regression Model to predict whether a lead for online courses for an education company named X Education would be successfully converted or not.*

Group Members

Sowjanya

Sohan

Nivedita

Business Objective

- *To help X Education to select the most promising leads(Hot Leads), i.e. the leads that are most likely to convert into paying customers.*
- *To build a logistic regression model to assign a lead score value between 0 and 100 to each of the leads which can be used by the company to target potential leads.*

The objective is thus classified into the following sub-goals:

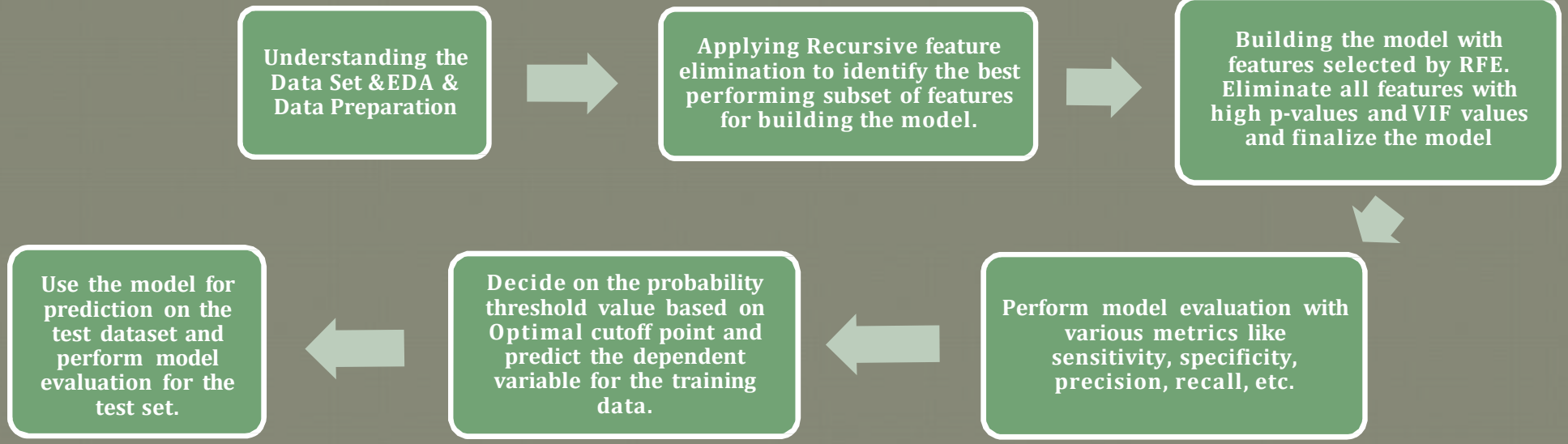
Create a Logistic Regression model to predict the Lead Conversion probabilities for each lead.

Decide on a probality threshold value above which a lead will be predicted as converted, whereas not converted if it is below it.

Multiply the Lead Conversion probability to arrive at the Lead Score value for each lead.

Problem Solving Methodology

- The approach for this project has been to divide the entire case study into various checkpoints to meet each of the sub-goals. The checkpoints are represented in a sequential flow as below:



Data Preparation & Feature Engineering

The following data preparation processes were applied to make the data dependable so that it can provide significant business value by improving Decision Making Process:

Remove columns which has only one unique value	• Deleting the following columns as they have only one unique value and hence cannot be responsible in predicting a successful lead case – **'Magazine','Receive More Updates About Our Courses','I agree to pay the amount through cheque','Get updates on DM Content','Update me on Supply Chain Content'.**
Removing rows where a particular column has high missing values	• **'Lead Source'** is an important column for analysis. Hence all the rows that have null values for it were dropped.
Imputing NULL values with Median	• The columns **'TotalVisits'** and **'Page Views Per Visit'** are continuous variables with outliers. Hence the null values for these columns were imputed with the column median values.
Imputing NULL values with Mode	• The columns **'Country'** is a categorical variable with some null values. Also majority of the records belong to the Country 'India'. Thus imputed the null values for this with mode(most occuring value). Then binned rest of category into 'Outside India'.

Data Preparation & Feature Engineering contd…

Handling 'Select' values in some columns

- There are some columns in dataset which have a level/value called 'Select'. This might have happened because these fields in the website might be non mandatory fields with drop downs options for the customer to choose from. Amongst the dropdown values, the default option is probably 'Select' and since these aren't mandatory fields, many customer might have have chosen to leave it as the default value 'Select'.
- The Select values in columns were **converted to Nulls**.

Assigning a Unique Category to NULL/SELECT values

- All the nulls in the columns were binned into a separate column '**not provided**'.
- Instead of deleting columns with huge null value percentage(which results in loss of data), this strategy adds more information into the dataset and results in the change of variance.
- The Unknown levels for each of these columns will be finally dropped during dummy encoding.

Outlier Treatment

- The outliers present in the columns '**TotalVisits**' & '**Page Views Per Visit**' were finally removed based on iterquatile range analysis.

Binary Encoding

- Converting the following binary variables (Yes/No) to 0/1:
- **'A free copy of Mastering The Interview', 'Do Not Email'**

Data Preparation & Feature Engineering contd…

Dummy Encoding

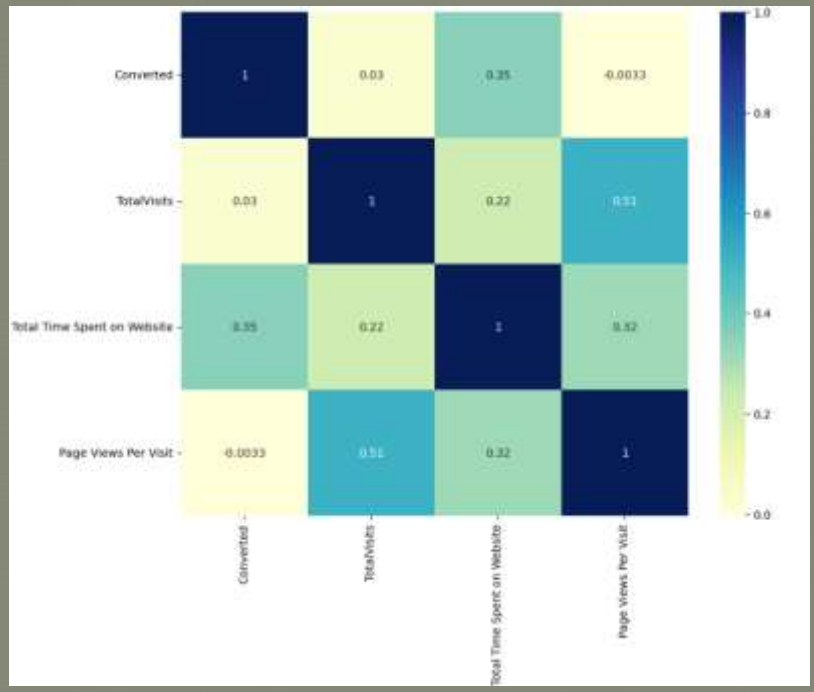
- For the following categorical variables with multiple levels, dummy features (one-hot encoded) were created:
- **'Lead Origin','What is your current occupation','City', 'Specialization', 'Lead Source', 'Last Activity','Last Notable Activity', 'Tags'.**

Test-Train Split

- The original dataframe was split into **train and test** dataset. The train dataset was used to train the model and test dataset was used to evaluate the model.

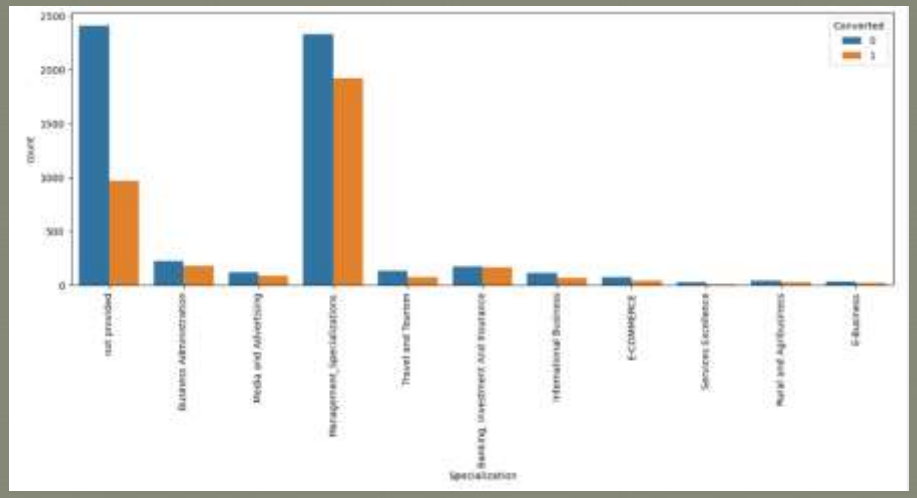
Feature Scaling

- Scaling helps in interpretation. It is important to have all variables(specially categorical ones which has values 0 and 1) on the same scale for the model to be easily interpretable.
- '**Standardisation**' was used to scale the data for modelling. It basically brings all of the data into a standard normal distribution with mean at zero and standard deviation one.

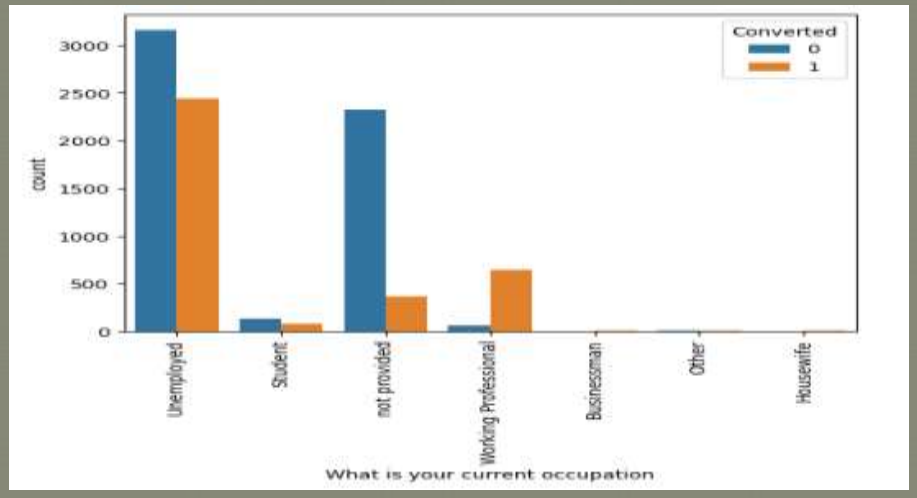


Heat map of numeric features.

Exploratory Data Analysis

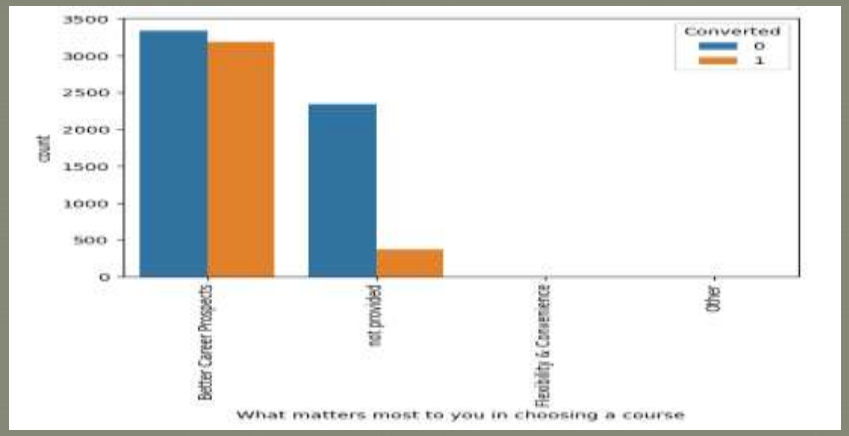


Specialization vs Converted

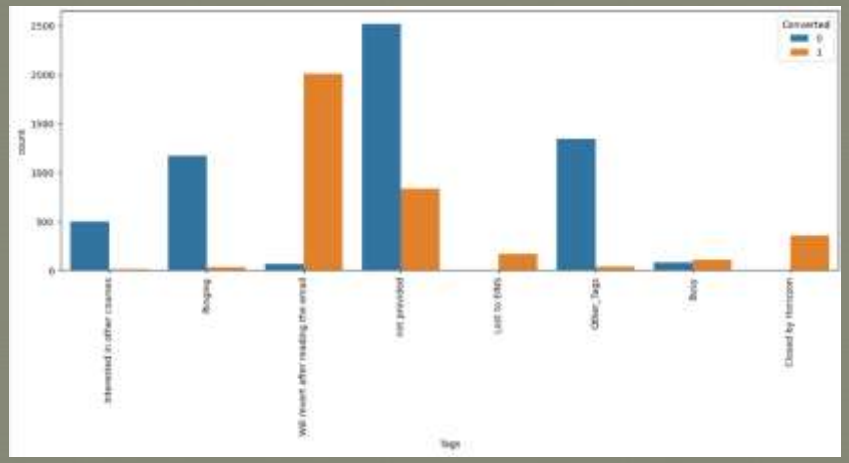


What is your current occupation vs converted

Exploratory Data Analysis Contd…

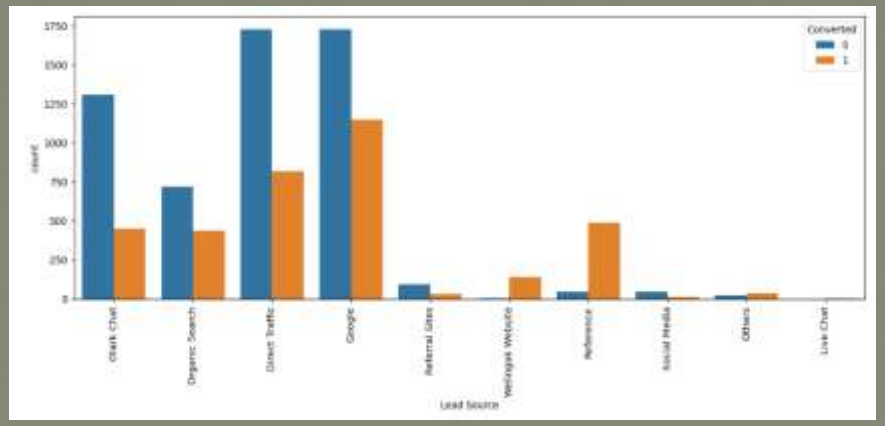


What matters most to you in choosing a course
vs Converted

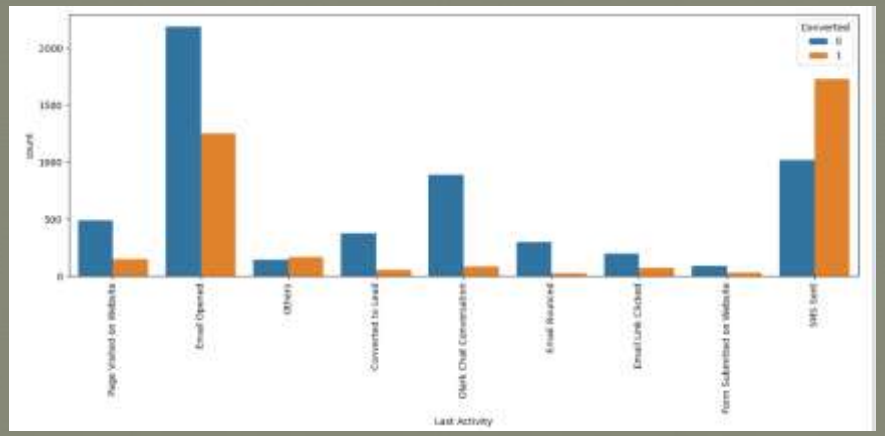


Tags vs converted

Exploratory Data Analysis Contd…

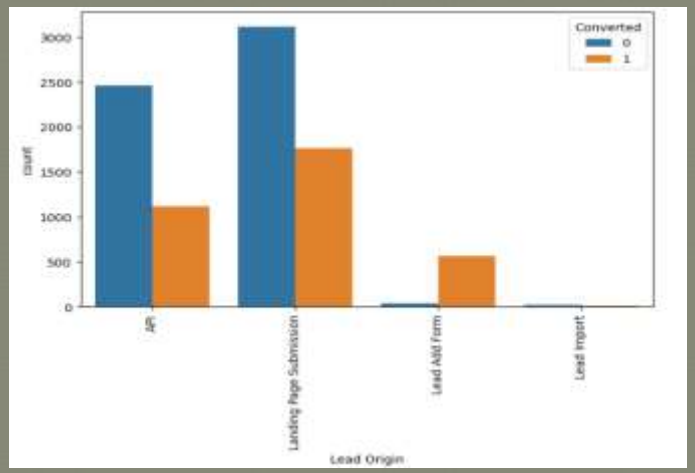


Lead Source vs Converted

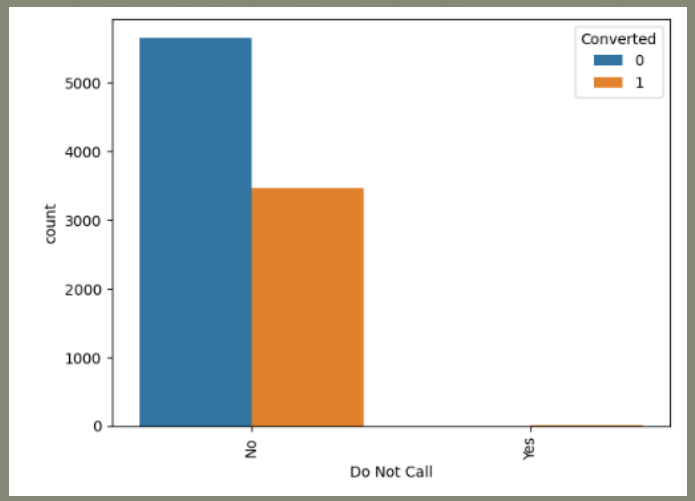


Last Activity vs converted

Exploratory Data Analysis Contd…

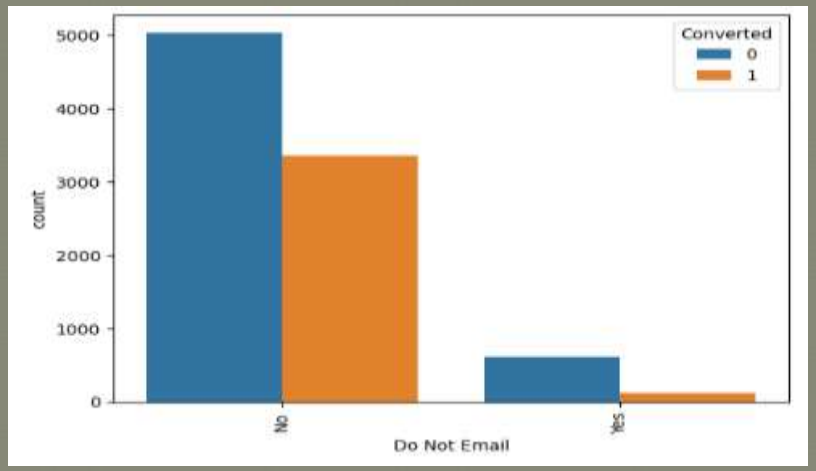


Lead Origin vs Converted

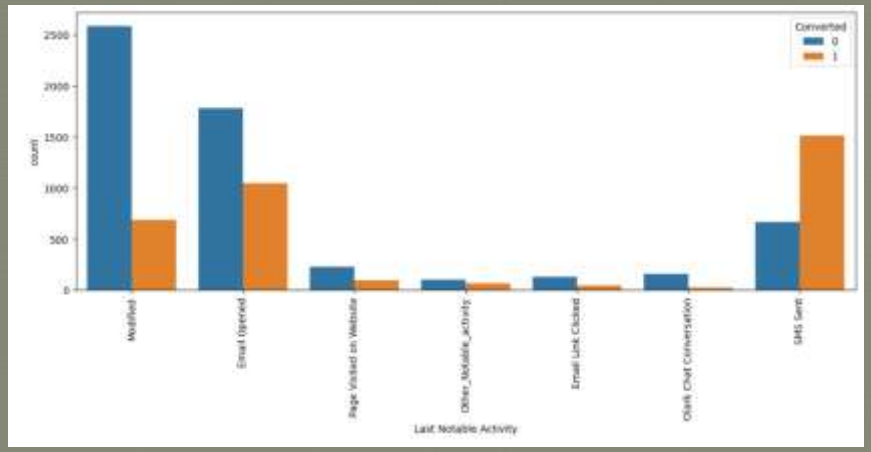


Do not Call vs Converted

Exploratory Data Analysis Contd…

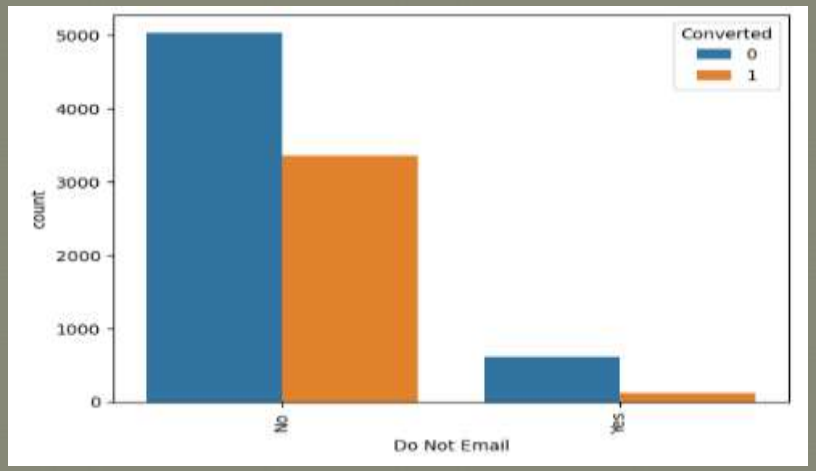


Do not Email vs Converted

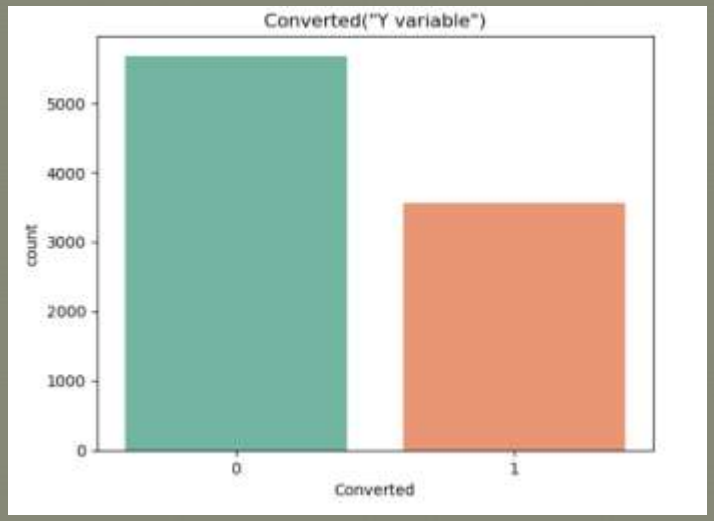


Last notable Activity vs Converted

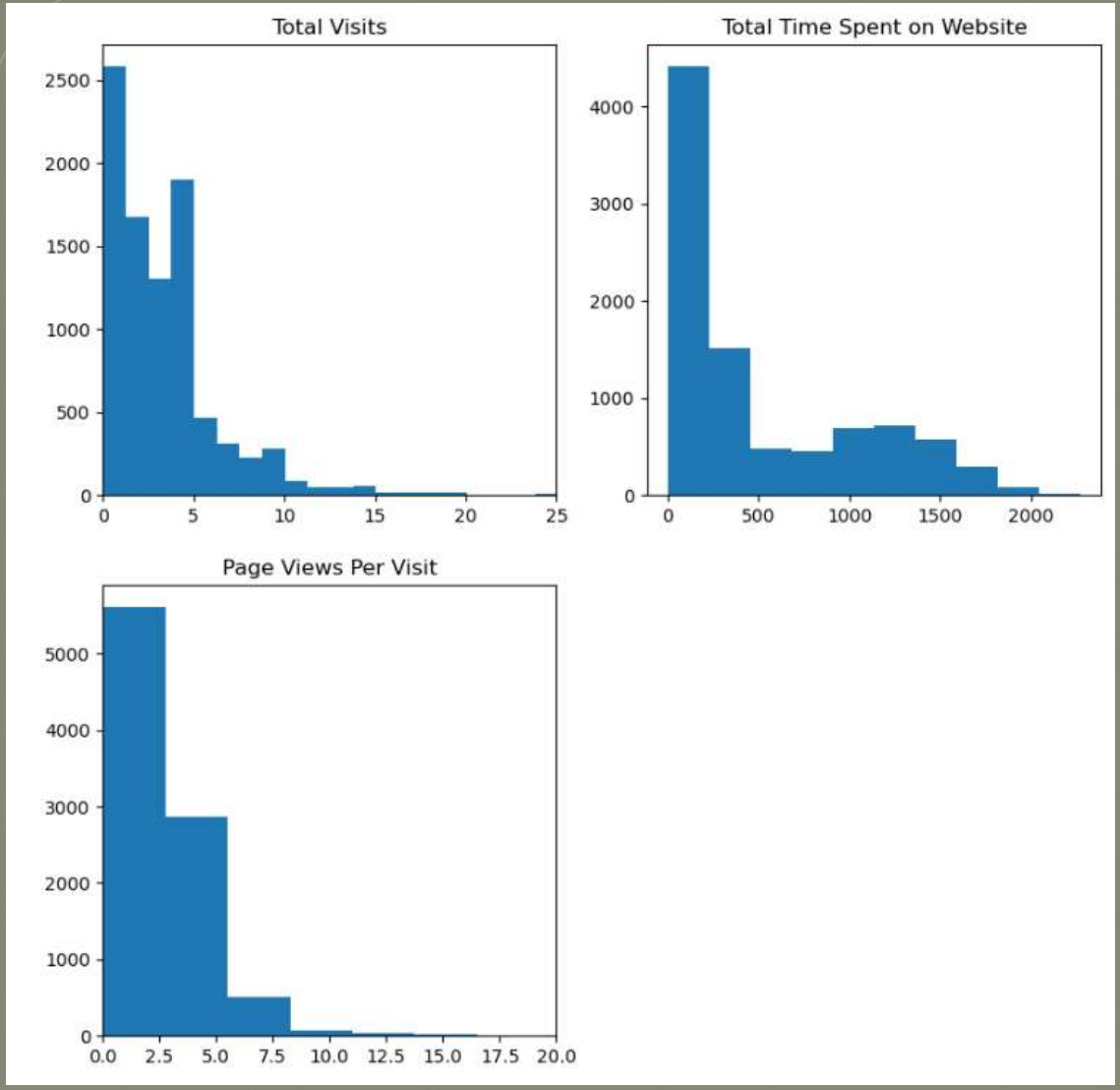
Exploratory Data Analysis Contd…



Do not Email vs Converted



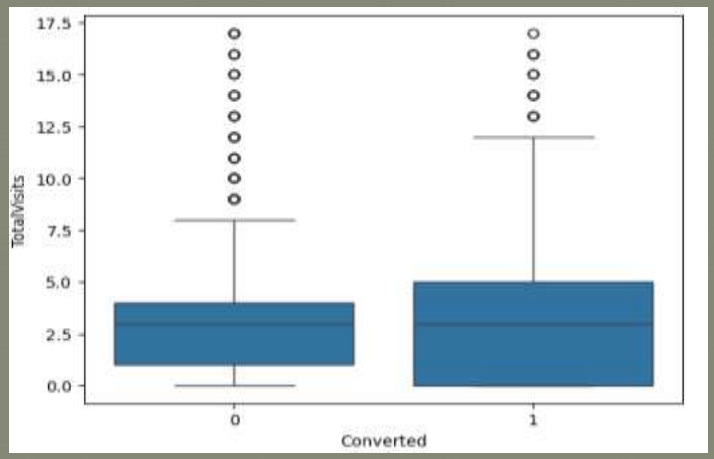
Conversion Rate



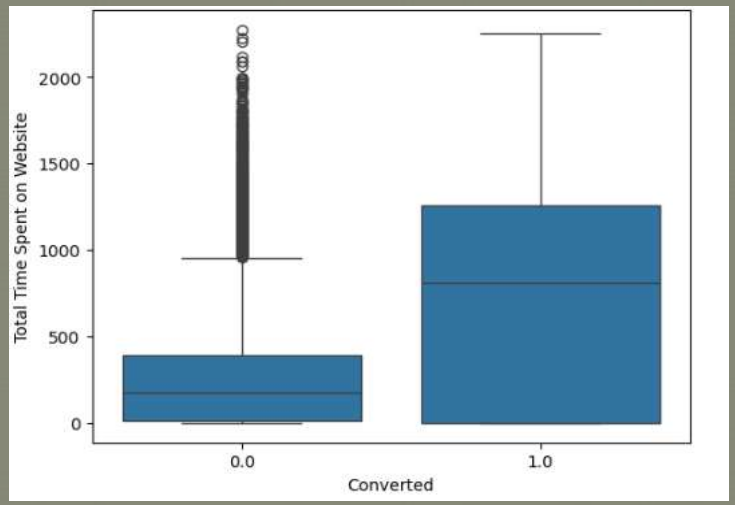
Exploratory Data Analysis Contd…

- Total Visits
  Total Time Spent on Website
- Page Views Per Visit

Exploratory Data Analysis Contd…

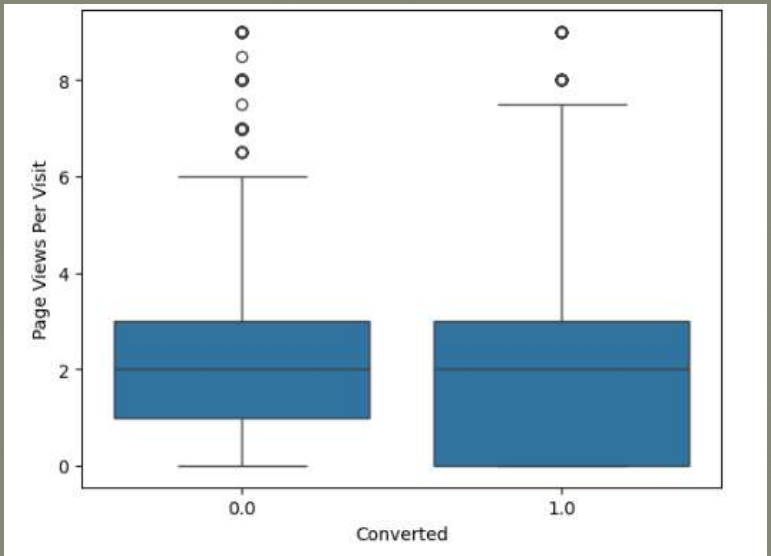


Total Visits vs Converted



Total Time Spent on Website Vs Converted

Exploratory Data Analysis Contd…



Page views per Visit vs Converted

Feature Selection Using RFE

•**Recursive feature elimination** is an optimization technique for finding the best performing subset of features. It is based on the idea of repeatedly constructing a model and choosing either the best (based on coefficients), setting the feature aside and then repeating the process with the rest of the features. This process is applied until all the features in the dataset are exhausted. Features are then ranked according to when they were eliminated.

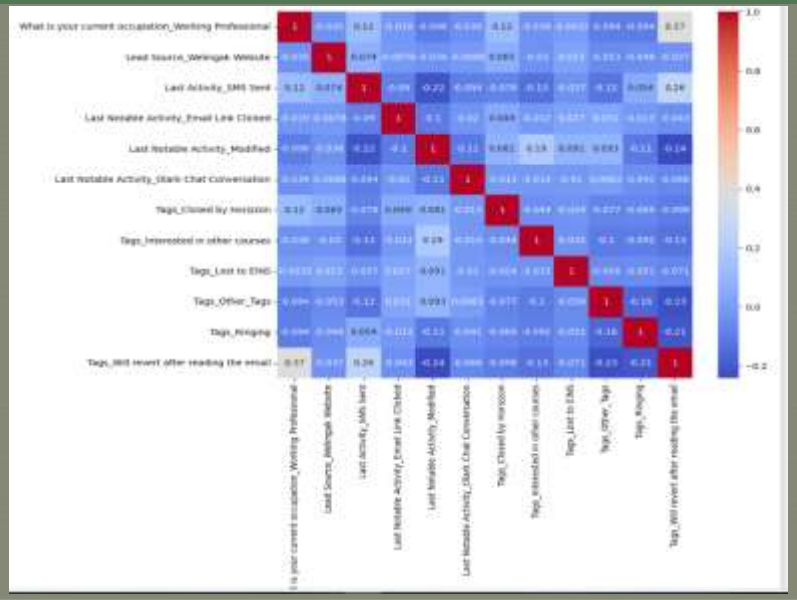
```
#list of RFE supported columns
col = X_train.columns[rfe.support_]
col
```

```
Index(['Lead Origin_Lead Add Form', 'What is your current occupation_Student',
       'What is your current occupation_Unemployed',
       'What is your current occupation_Working Professional',
       'Lead Source_Welingak Website', 'Last Activity_SMS Sent',
       'Last Notable Activity_Email Link Clicked',
       'Last Notable Activity_Modified',
       'Last Notable Activity_Olark Chat Conversation',
       'Tags_Closed by Horizzon', 'Tags_Interested in other courses',
       'Tags_Lost to EINS', 'Tags_Other_Tags', 'Tags_Ringing',
       'Tags_Will revert after reading the email'],
      dtype='object')
```

•**Running RFE with the output number of the variable equal to 15.**

Building the model

- Generalized Linear Models from StatsModels is used to build the Logistic Regression model.
- The model is built initially with the 15 variables selected by RFE.
- Unwanted features are dropped serially after checking p values (< 0.5) and VIF (< 5) and model is built multiple times.
- The final model with 16 features, passes both the significance test and the multi-collinearity test.



	Features	VIF
11	Tags_Will revert after reading the email	1.53
2	Last Activity_SMS Sent	1.42
4	Last Notable Activity_Modified	1.36
0	What is your current occupation_Working Profes...	1.30
9	Tags_Other_Tags	1.16
7	Tags_Interested in other courses	1.12
10	Tags_Ringing	1.10
6	Tags_Closed by Horizzon	1.09
8	Tags_Lost to EINS	1.04
1	Lead Source_Welingak Website	1.03
3	Last Notable Activity_Email Link Clicked	1.02
5	Last Notable Activity_Olark Chat Conversation	1.01

	coef	std err	z	P>|z|
const	-1.2351	0.075	-16.437	0.000
What is your current occupation_Working Professional	0.9524	0.404	2.356	0.018
Lead Source_Welingak Website	5.0740	1.026	4.945	0.000
Last Activity_SMS Sent	2.0657	0.106	19.527	0.000
Last Notable Activity_Email Link Clicked	-1.5578	0.487	-3.197	0.001
Last Notable Activity_Modified	-1.7665	0.119	-14.841	0.000
Last Notable Activity_Olark Chat Conversation	-1.6853	0.451	-3.740	0.000
Tags_Closed by Horizzon	7.5780	1.011	7.498	0.000
Tags_Interested in other courses	-1.8194	0.375	-4.847	0.000
Tags_Lost to EINS	6.0231	0.599	10.050	0.000
Tags_Other_Tags	-2.4013	0.196	-12.272	0.000
Tags_Ringing	-3.3455	0.225	-14.872	0.000
Tags_Will revert after reading the email	4.5702	0.183	24.978	0.000

- A heat map consisting of the final 12 features proves that there is no significant correlation between the independent variables.

Predicting the Conversion Probability and Predicted column

Creating a dataframe with the actual Converted flag and the predicted probabilities.

Showing top 5 records of the dataframe in the picture on the right.

	Converted	Converted_prob	Prospect ID
0	1	0.074819	9196
1	0	0.074819	4696
2	0	0.696483	3274
3	0	0.007994	2164
4	1	0.965612	1667

	Converted	Converted_prob	Prospect ID	Predicted
0	1	0.074819	9196	0
1	0	0.074819	4696	0
2	0	0.696483	3274	1
3	0	0.007994	2164	0
4	1	0.965612	1667	1

Creating new column 'predicted' with 1 if Conversion_Prob > 0.5 else 0

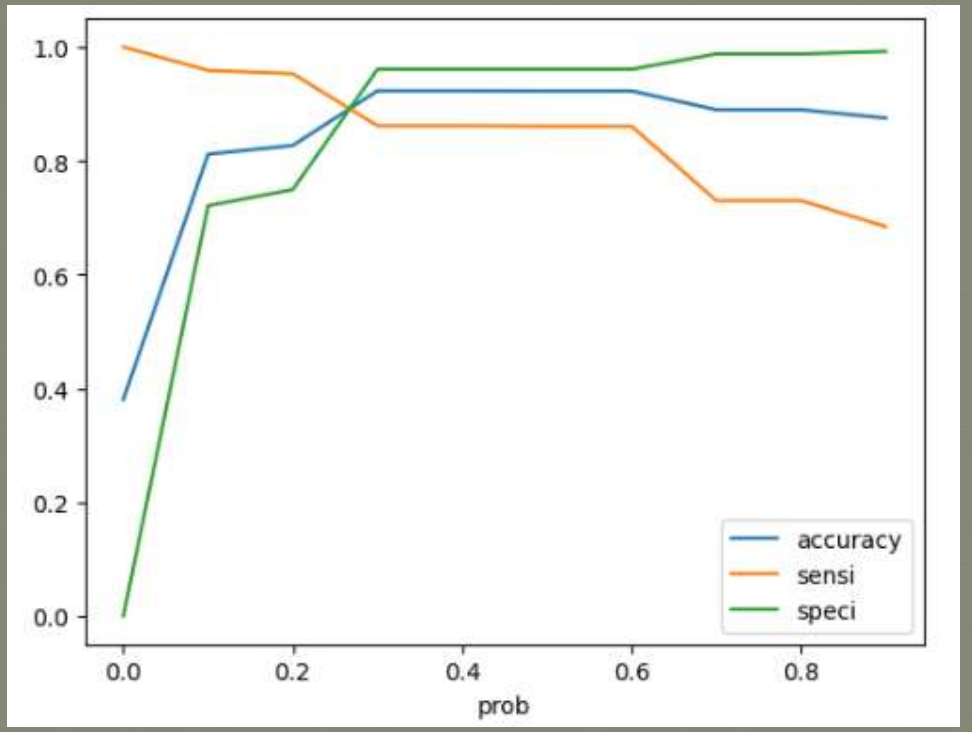
Showing top 5 records of the dataframe in the picture on the left.

Finding Optimal Probability Threshold

Optimal cutoff probability is that prob where we get balanced sensitivity and specificity.

Optimal Probability Threshold

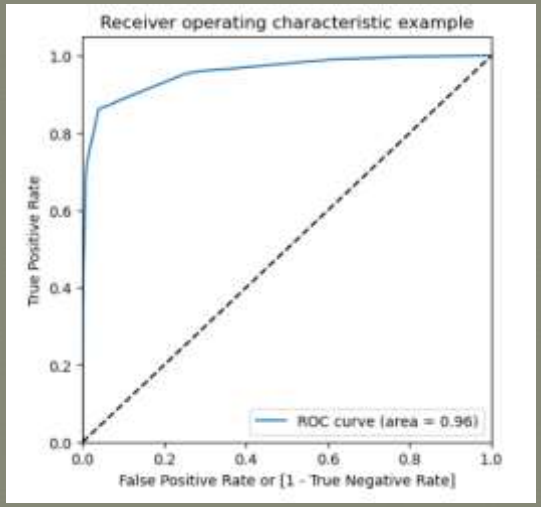
- The accuracy sensitivity and specificity was calculated for various values of probability threshold and plotted in the graph to the right.
- From the curve above, **0.3** is found to be the optimum point for cutoff probability.
- At this threshold value, all the 3 metrics - accuracy sensitivity and specificity was found to be well above 80% which is a well acceptable value.



Plotting the ROC Curve & Calculating AUC

Receiver Operating Characteristics (ROC) Curve

- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).



Area under the Curve (GINI)

- By determining the Area under the curve (AUC) of the ROC curve, the goodness of the model is determined. Since the ROC curve is more towards the upper-left corner of the graph, it means that the model is very good. The larger the AUC, the better will is the model.
- The value of AUC for our model is 0.96.

As a rule of thumb, an AUC can be classed as follows,

- 0.90 - 1.00 = excellent
- 0.80 - 0.90 = good
- 0.70 - 0.80 = fair
- 0.60 - 0.70 = poor
- 0.50 - 0.60 = fail

Since we got a value of 0.9678, our model seems to be doing well on the test dataset.

Evaluating the model on train datatset

Confusion Matrix

# Predicted	Not Converted	Converted
# Actual		
Not Converted	3730	152
Converted	330	2055

Probability Threshold = 0.3

Accuracy TP +TN/ (TP+TN+FN+FP)	Sensitivity TP / (TP+FN)	Specificity TN / (TN+FP)	False Positive Rate FP/ (TN+FP)	Positive Predictive Value TP / (TP+FP)
0.923	0.861	0.960	0.039	0.931

Negative Predictive Value TN / (TN+ FN)	Precision TP / TP + FP	Recall TP / TP + FN	*F1 score = 2×(Precision*Recall)/(Precision+Recall)*	Area under the cuve
0.918	0.931	0.861	0.894	0.96

Making predictions on the test set

- The final model on the train dataset is used to make predictions for the test dataset
- The train data set was scaled using the scaler.transform function that was used to scale the train dataset.
- The Predicted probabilities were added to the leads in the test dataframe.
- Using the probability threshold value of 0.3, the leads from the test dataset were predicted if they will convert or not.

- The Conversion Matrix was calculated based on the Actual and Predicted 'Converted' columns.

The top 5 records from the final test data set

	Prospect ID	Converted	Converted_prob	Lead_Score	final_Predicted
0	7681	0	0.074819	7	0
1	984	0	0.034316	3	0
2	8135	0	0.696483	70	1
3	6915	0	0.010144	1	0
4	2712	1	0.965612	97	1

Evaluating the model on test datatset

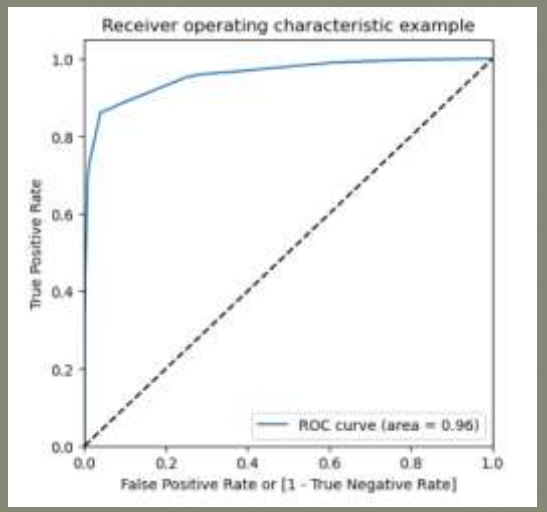
The following evaluation metrices were recorded for the test dataset.

Accuracy TP +TN/ (TP+TN+FN+FP)	Sensitivity TP / (TP+FN)	Specificity TN / (TN+FP)	Area under the cuve
0.926	0.869	0.961	0.96

Negative Predictive Value TN / (TN+ FN)	Precision TP / TP + FP	Recall TP / TP + FN	*F1 score = 2×(Precision*Recall)/(Precision+ Recall)*
0.924	0.931	0.869	0.898

False Positive Rate FP/ (TN+FP)	Positive Predictive Value TP / (TP+FP)
0.038	0.931

Area under the Curve



Lead Score Calculation

Lead Score is calculated for all the leads in the original dataframe.

Formula for Lead Score calculation is:

Lead Score = 100 * Conversion Probability

LeadID	Lead Number	Converted	Conversion_Prob	final_predicted	Lead_Score
0	660737	0	0.03	0	3
1	660728	0	0.01	0	1
2	660727	1	0.80	1	80
3	660719	0	0.01	0	1
4	660681	1	0.96	1	96
5	660680	0	0.08	0	8
6	660673	1	0.96	1	96
7	660664	0	0.08	0	8
8	660624	0	0.08	0	8
9	660616	0	0.08	0	8

The figure showing Lead Score for head and tail records from the data set.

- The train and test dataset is concatenated to get the entire list of leads available.
- The Conversion Probability is multiplied by 100 to obtain the Lead Score for each lead.
- Higher the lead score, higher is the probability of a lead getting converted and vice versa,
- Since, we had used 0.3 as our final Probability threshold for deciding if a lead will convert or not, any lead with a lead score of 30 or above will have a value of '1' in the final_predicted column.

Determining Feature Importance

- 16 features have been used by our model to successfully predict if a lead will get converted or not.
- The Coefficient (beta) values for each of these features from the model parameters are used to determine the order of importance of these features.
- Features with high positive beta values are the ones that contribute most towards the probability of a lead getting converted.
- Similarly, features with high negative beta values contribute the least.

Tags_Closed by Horizzon	7.578018
Tags_Lost to EINS	6.023126
Lead Source_Welingak Website	5.073982
Tags_Will revert after reading the email	4.570174
Last Activity_SMS Sent	2.065738
What is your current occupation_Working Professional	0.952375
const	-1.235130
Last Notable Activity_Email Link Clicked	-1.557804
Last Notable Activity_Olark Chat Conversation	-1.685278
Last Notable Activity_Modified	-1.766549
Tags_Interested in other courses	-1.819363
Tags_Other_Tags	-2.401273
Tags_Ringing	-3.345528
dtype: float64	

After trying several models, we finally chose a model with the following characteristics:

All variables have p-value < 0.05.

All the features have very low VIF values, meaning, there is hardly any multicollinearity among the features. This is also evident from the heat map.

The overall accuracy of 0.92 at a probability threshold of 0.3 on the test dataset is also very acceptable.

Inference contd...

Based on our model, some features are identified which contribute most to a Lead getting converted successfully.

The conversion probability of a lead increases with increase in values of the positive corelated features in descending order:

Tags_Closed by Horizzon	7.578018
Tags_Lost to EINS	6.023126
Lead Source_Welingak Website	5.073982
Tags_Will revert after reading the email	4.570174
Last Activity_SMS Sent	2.065738
What is your current occupation_Working Professional	0.952375

The conversion probability of a lead increases with decrease in values of the negative corelated features in descending order:

const	-1.235130
Last Notable Activity_Email Link Clicked	-1.557804
Last Notable Activity_Olark Chat Conversation	-1.685278
Last Notable Activity_Modified	-1.766549
Tags_Interested in other courses	-1.819363
Tags_Other_Tags	-2.401273
Tags_Ringing	-3.345528

Recommendations & Problem Solution

Which are the top three variables in your model that contribute most towards the probability of a lead getting converted?	• **Tags_Lost to EINS** • **Tags_Closed by Horizzon** • **Lead Source_Welingak Website**
What are the top 3 categorical/dummy variables in the model which get maximum focus in order to increase the probability of lead conversion?	• **Tags_Lost to EINS** • **Tags_Closed by Horizzon** • **Lead Source_Welingak Website**
X Education has a period of 2 months every year during which they hire few interns.The sales team, in particular, has around 10 interns allotted to them. So, during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.	• **We will choose a lower threshold value for Conversion Probability. This will ensure the Sensitivity rating is very high which in turn will make sure almost all leads that are likely to Convert are identified correctly and the agents can make phone calls to as much of such people as possible.**
Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.	• **We will choose a higher threshold value for Conversion Probability. This will ensure the Specificity rating is very high, which in turn will make sure almost all leads that are on the brink of the probability of getting Converted or not are not selected. As a result the agents won't have to make unnecessary phone calls and can focus on some new work.**