LEAD SCORING CASE STUDY

SUBMITTED BY:
SHARAD CHOUDHURY
KALYANI BURANGE
MOHIT KOSEKAR

PROBLEM STATEMENT

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.

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.

Some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.

The company requires us to build a model wherein we 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, in particular, has given a ballpark of the target lead conversion rate to be around 80%.

SOLUTION APPROACH

Step 1: Importing and analyzing data.

- Import the data and analyze its data types.
- Find percentage of missing values in each column.
- Find the spread of data in the numerical columns.
- Find the count of unique categories in each column.
- The dataset has 9240 rows and 37 columns.

Step 2: Data Cleaning

- Replace ‘Select’ category with Nulls.
- Remove highly skewed and unnecessary columns :
  - Prospect ID , Lead Number are irrelevant for prediction as they are unique IDs.
  - There are also many skewed columns which have high frequency of data in only one category and other categories have negligible frequency.
    - Eg: 'Do Not Call', 'Country’, 'Search', 'Magazine', 'Newspaper Article’ etc.
- We drop these columns too.

- Remove columns with >= 30 % null values as columns with greater than 30% null values are irrelevant for prediction and imputing them might create skewness affecting the model performance.
- Impute the nulls in those columns where null percentage is around 1%.
- Clean the categorical columns by removing duplicates and grouping low frequency(counts below 1%) categories into one.
- Handle the outliers by capping them.

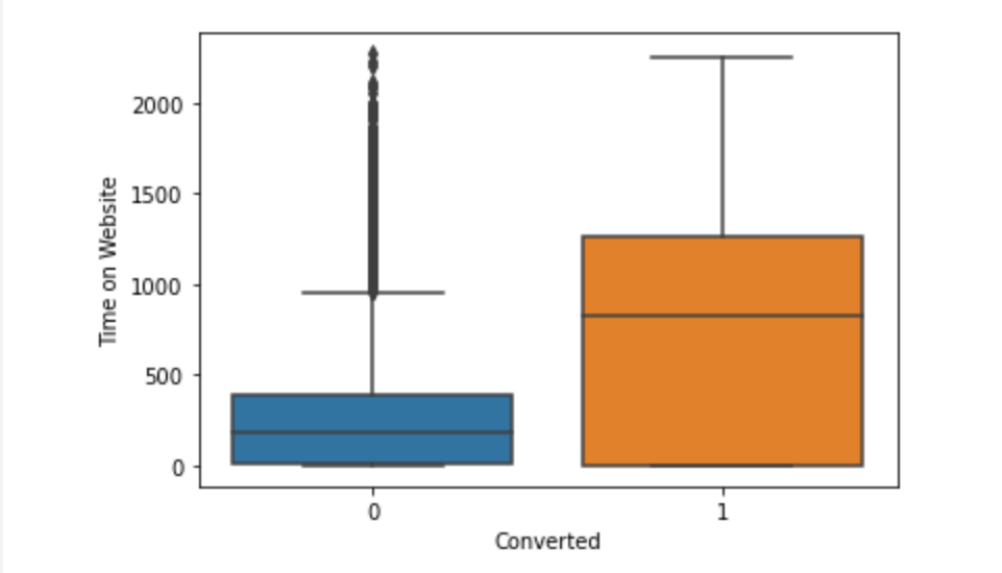
Step 3: EDA

- Performed EDA on categorical variables

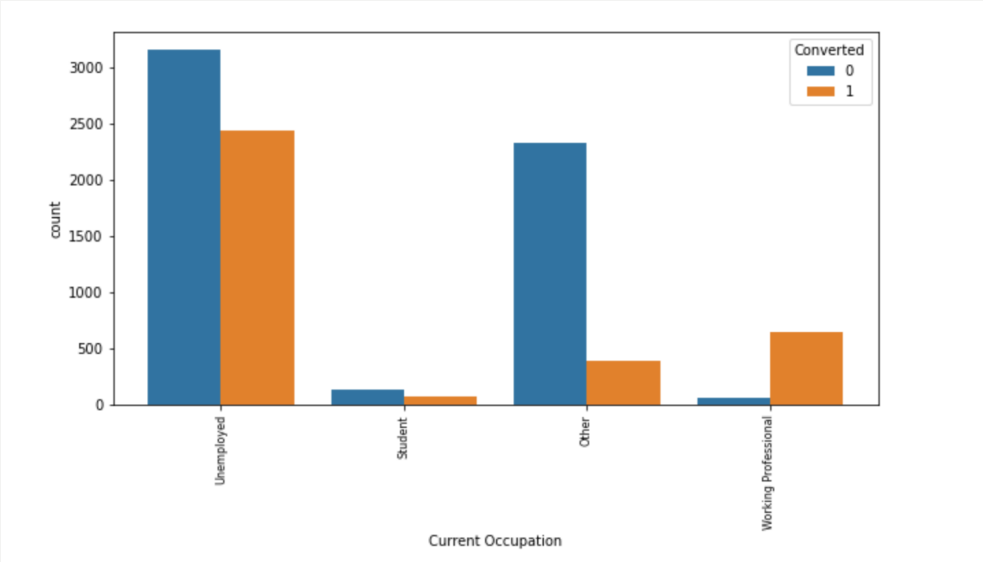
to find their category-wise counts wrt Conversion type.

- Plotted the boxplots of numerical attributes wrt

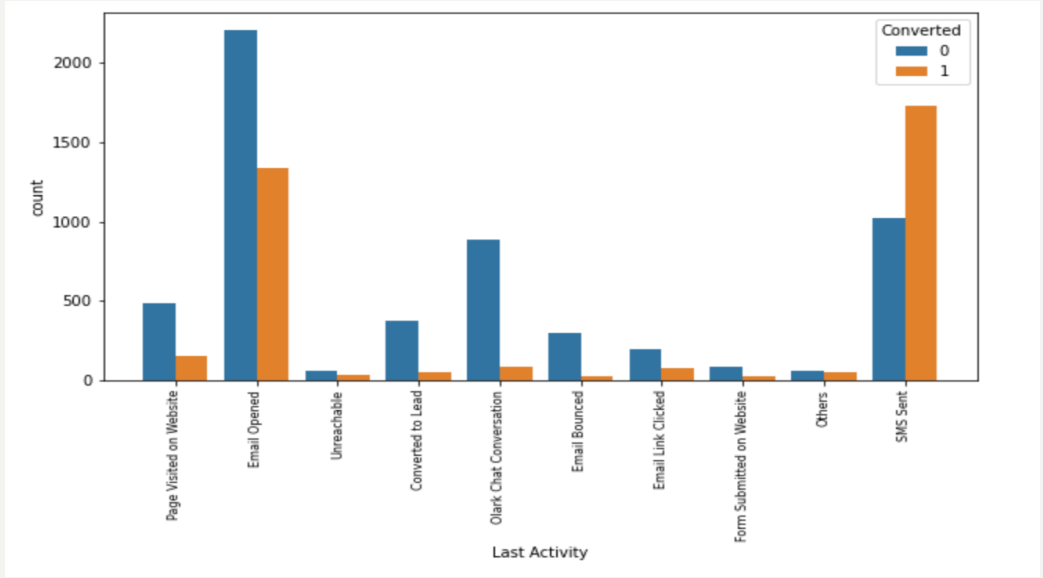
Converted type.



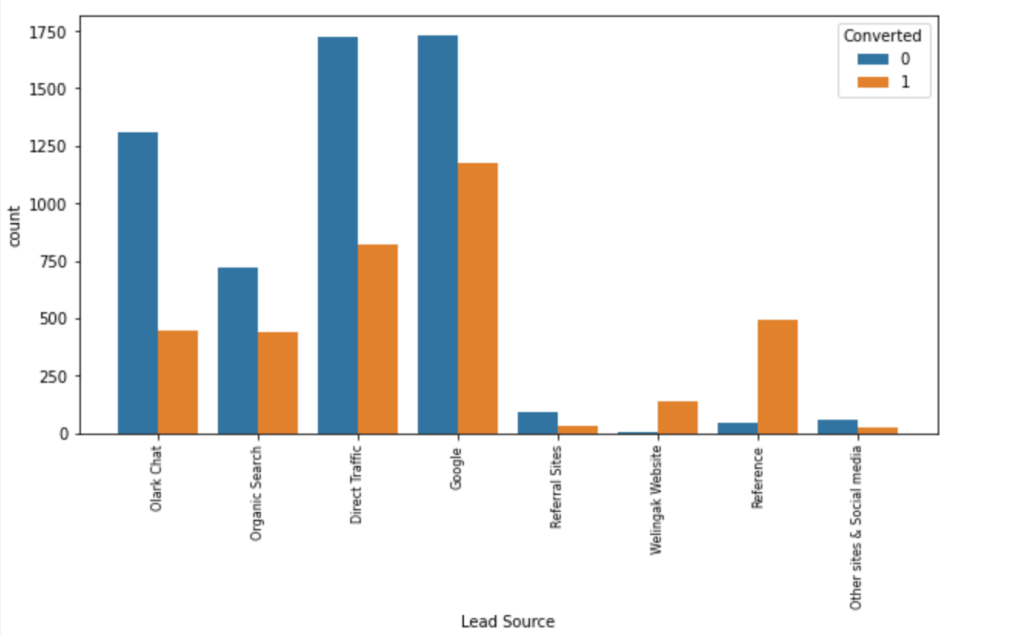
Leads who spent more time on website had higher conversion rates.



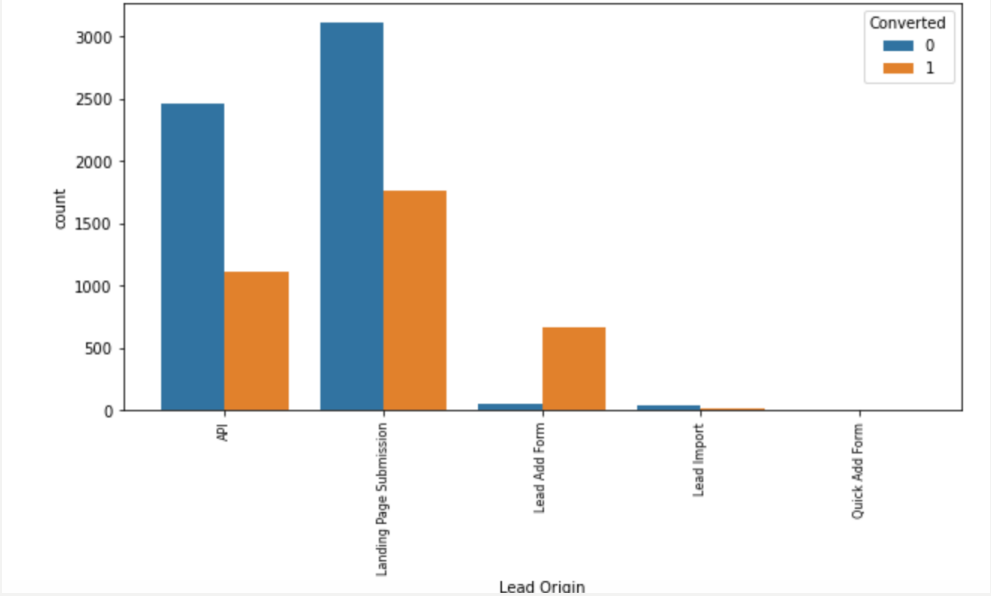
Working Professionals had highest conversion rate



Leads with SMS sent as Last activity had highest conversion rate



Leads through reference had highest conversion rate



Leads with origin as Lead Add form had the highest conversion rate

Step 4: Data Preparation

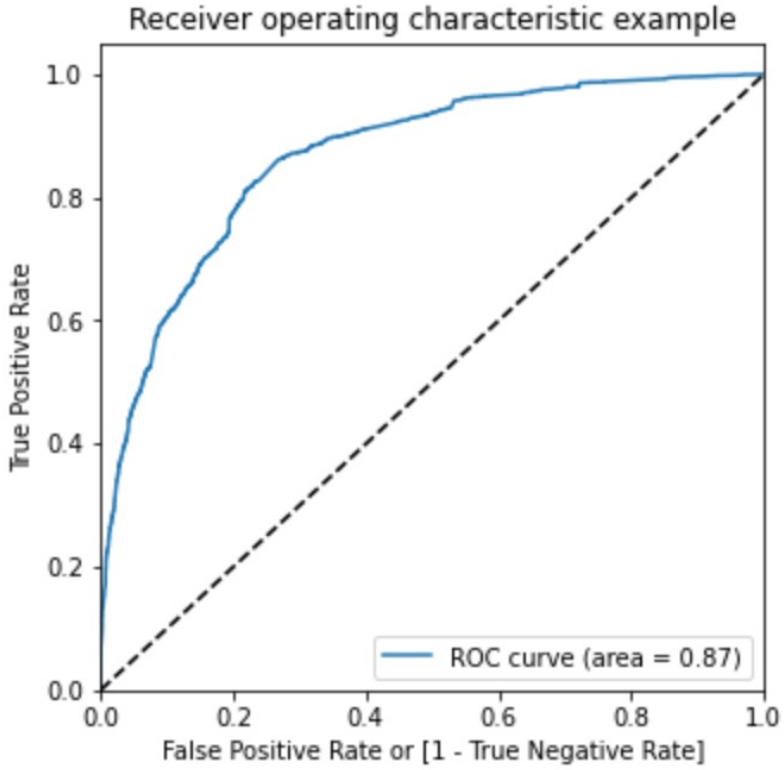
- Create dummies for the categorical columns.
- Divide into train and test sets in 70:30 ratio.
- Scale the numerical columns in train set using MinMaxScaler. We need to scale the data because we will be using a Logistic regression model for prediction and if we don't scale the continuous numerical columns then the coefficients will be higher for higher range values thus reducing the model interpretability.

Step 5: Model Building

- Initialize Logistic Regression model.
- Perform coarse feature selection using Recursive Feature Elimination and select top 15 features.
- Perform fine feature selection by removing the features with high p-values and Variance Inflation Factor (VIF).
- In the end, we had 11 features in the final model and all of them had a p-value < 0.05 and VIF < 5.

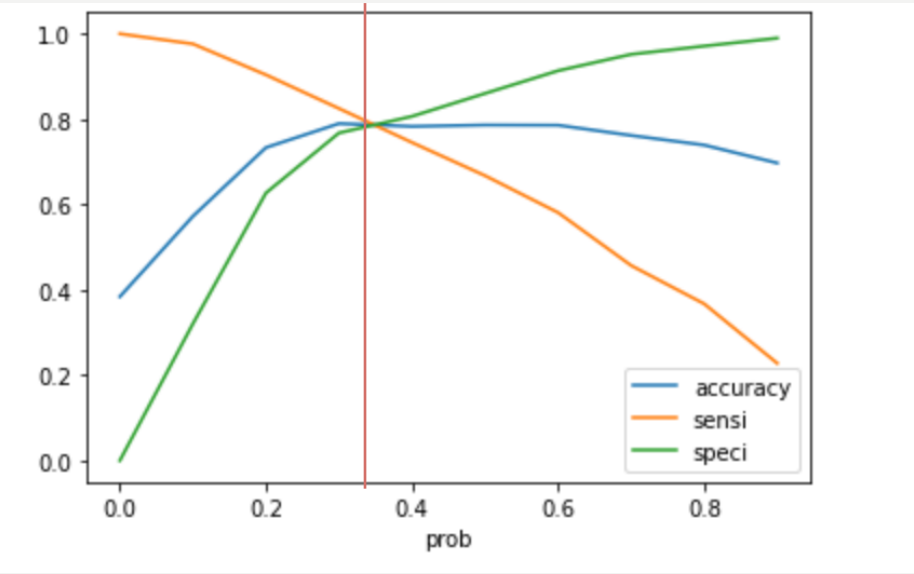
Step 6: Model Performance Evaluation

- Make predictions using the final model on the train set.
- Plot the ROC curve for the model. We obtained an impressive ROC curve that aligned to the top left edge of the plot and AUC = 0.87.
- Using the confusion matrix, plot the Accuracy, Sensitivity, Specificity for the model at different probability thresholds.
- Sensitivity or Recall is the Total number of Leads correctly predicted as Converted out of the Total number of actual Converted Leads.
- $$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{Total Actual Positives (TP + FN)}}$$
- Precision is the Total number of leads correctly predicted as Converted out of the total number of leads which are predicted as Converted.
- $$\text{Precision} = \frac{\text{True positives (TP)}}{\text{Total Predicted positives (TP+FP)}}$$
- In our case, if our precision is a bit low, we will end up reaching to few non potential leads too. So, we can tradeoff a little precision for higher recall.

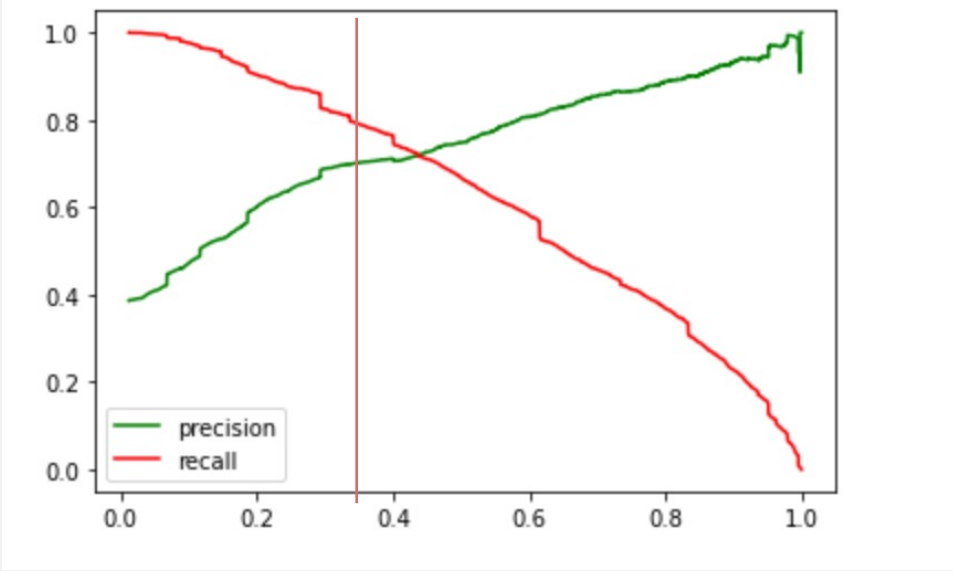


ROC curve of the model

- Our goal is to reach out to as many leads as possible to ensure maximum conversion rate. So, our probability cutoff should be such that the Recall will increase so that we can identify most of the potential leads (Hot leads) and the Precision is also decent, which means minimise unnecessary calls to non potential leads.
- So, we set 0.35 as our Cut-off probability so we get a Higher Recall and fairly high Precision.
- We obtained a target lead conversion rate of around 79% with our model. This means that out of the total leads who actually got converted we were able to reach out to 79% of them to ensure high conversion rate.



Accuracy – Sensitivity – Specificity tradeoff



Precision – Recall tradeoff

The Red line indicates the probability cutoff above which all leads who be classified as Converted.

Step 7: Evaluation of model on the test set

- Transform the numerical columns in test set using the MinMaxScaler object.
- Choose those columns from test set that are used in the final prediction model in the train set.
- Find the predictions on the test set and evaluate the performance metrics.

	Accuracy	Precision	Recall	f1-score
Train	0.790816	0.70247	0.789855	0.743604
Test	0.797619	0.711823	0.805014	0.755556

Step 8: Forming the Scores dataframe

- Join the initial dataframe and predictions dataframe on 'ID' column which is actually the index of the rows from initial dataframe.
- Score = (Probability of Conversion) * 100
- Final scores dataframe has the Prospect ID, Lead Number, Actual Converted column, predicted Converted column and the Score.

	Prospect ID	Lead Number	Converted	final_predicted	Score
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	0	0	18.59
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	0	1	38.12
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	1	1	73.17