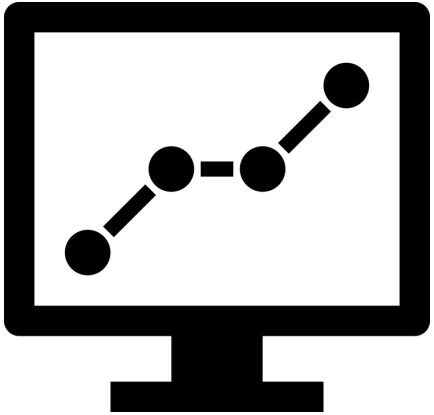
LEAD SCORING CASE STUDY



SUBMITTED BY:
PRACHI BANSAL
SOURAV KUMAR SINGH

PROBLEM STATEMENT

An education company provides online courses to industry professionals. X Education has appointed you to help them select the most promising leads, i.e. the leads that are most likely to convert into paying customers. 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.

AIM:

1. To develop a logistic regression model that assigns lead score from 1 to 100 to each of the leads
2. Predict whether the lead will convert to a customer or not based on the lead score/probability
3. Understand the model performance

WORKFLOW


1
DATA UNDERSTANDING
Importing the data through pandas and get to know the shape and basic information about its columns i.e data types, no. of nulls etc.
2
DATA PREPARATION
This includes multiple steps like data cleaning, outlier analysis, finding correlation between variables, splitting the data to train-test and feature scaling.
3
DATA MODELING
Here the model was fitted using logistic regression classifier and then i checked for p-values and VIF for each variable, found accuracy, sensitivity and specificity and optimal cutoff and made the predictions on train set.
3
PREDICTION ON TEST DATA
Now, predictions were made on the test data with the developed model and all the performance metrics were evaluated.

DATA PREPARATION

HANDLING MISSING VALUES:

There are 2 components to it:

1. Checking for null values
2. Checking for bad/dummy values

```
df["City"].value_counts()
```

```
Mumbai                          3165
Select                          2172
Thane & Outskirts                741
Other Cities                     675
Other Cities of Maharashtra      444
Other Metro Cities               377
Tier II Cities                    74
Name: City, dtype: int64
```

Null values: out of initial 37 columns, about 17 had nulls. % nulls for each of the columns were calculated and

1. The columns having null % ~>20-25% were dropped since its difficult to impute categorical variable.
2. The columns having null % < 2%, the records having nulls in those fields were deleted.

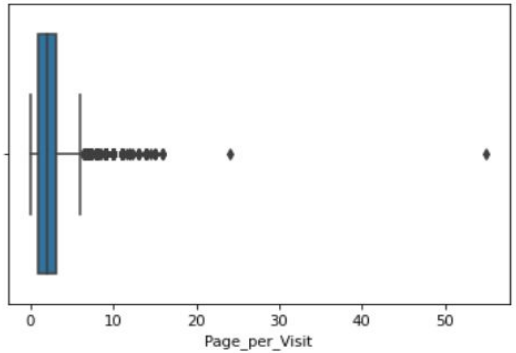
Bad/Dummy values:

1. Out of the remaining columns, 3 columns had dummy value “Select” and those columns were dropped.

Note: Some categorical columns that had just one category or having very few records of one category out of 2 categories were dropped since these columns did not provide significant information to our model.

OUTLIER ANALYSIS

1. For the numerical columns, outlier analysis was done
2. 3 numeric columns are there: "TotalVisits","Time_Website" and "Page_per_Visit"
3. Boxplots were plotted for each of the columns
4. TotalVisits and Page_per_Visit had outliers, and those records were removed

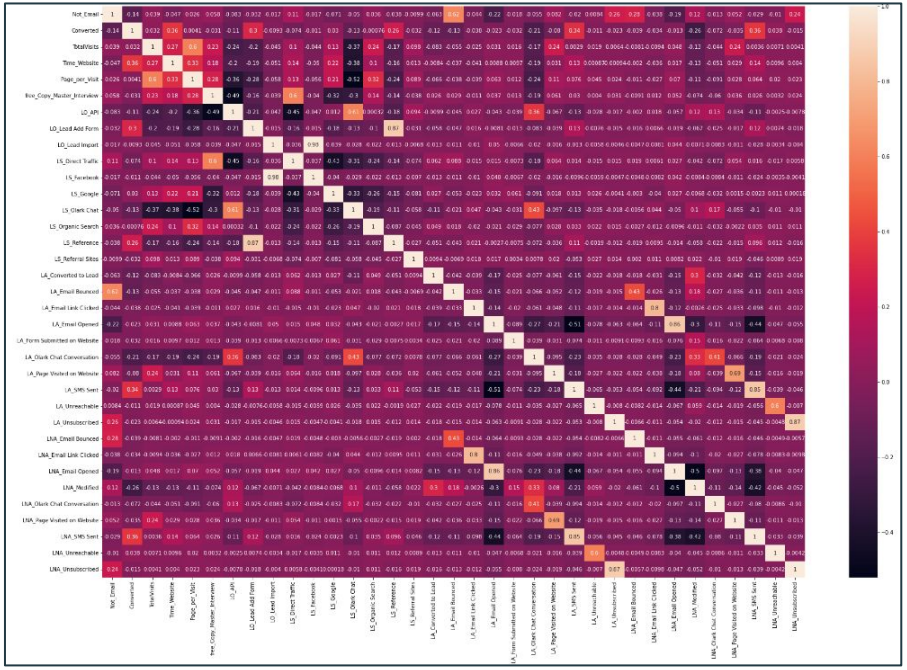


DUMMY VALUES CREATION FOR CATEGORICAL VARIABLES

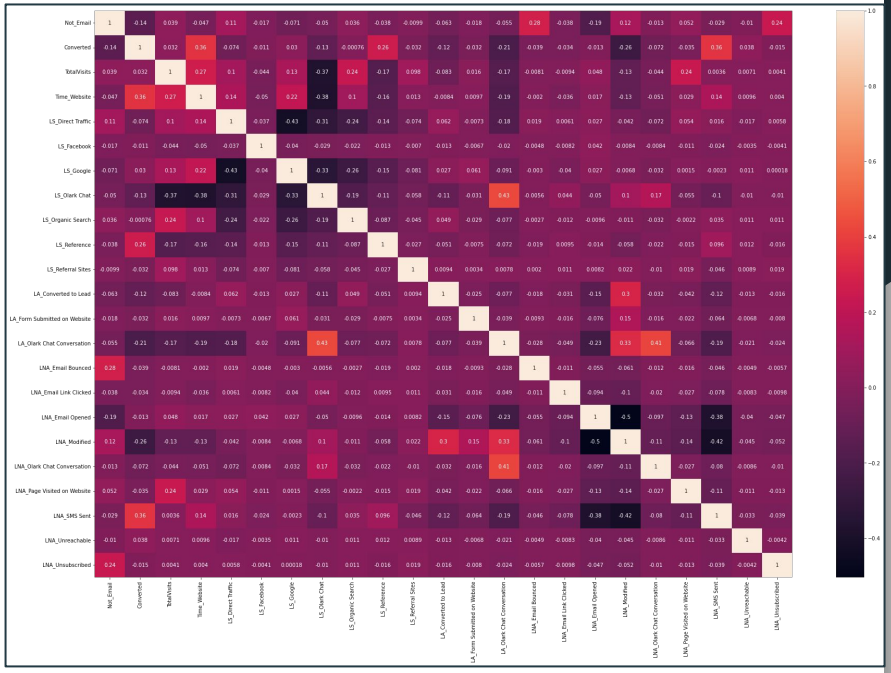
1. For binary categorical variables like Not_Email and free_Copy_Master_Interview, 0/1 are assigned for No/Yes
2. For categorical variables having more than 2 levels, dummy variables were created. Such columns were: Lead_Origin, Lead_Source, Last_Activity and Last_Notable_Activity.

CHECKING CORRELATION BETWEEN FEATURE VARIABLES

1. Plotted correlation matrix between all the variables and columns having high correlation values among them were identified.
2. Out of the pair of fields having high correlation, one was dropped from the dataframe.
3. Final correlation matrix was plotted and there was no significant correlation values observed between the variables.

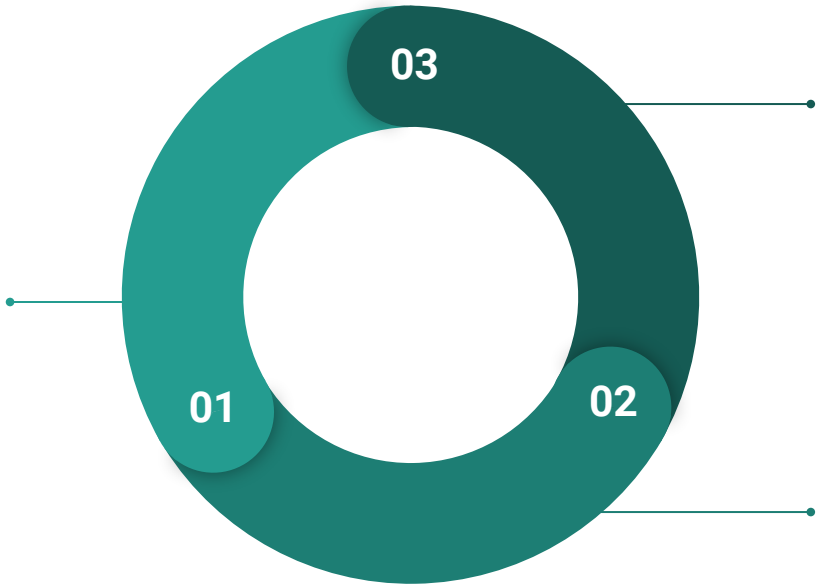


Correlation matrix with all the variables. Here the lighter shades indicate high correlation and the darker shades indicate low correlation values.



Here, we can see we don't have very high correlations present (no light shades)

TRAIN-TEST SPLIT
Train size taken to be 0.7 and random state 100.



CHECKING DATA IMBALANCE

Churn Rate came out to be ~ 38%, thus data is almost balanced and good for analysis

FEATURE SCALING

StandardScaler is used for this. Thus, on scaling we can get negative values as well

Finally after the data preparation steps, we had 9026 rows and 22 columns out of which 6318 was test records

DATA MODELING

MODEL FITTING USING ALL VARIBALES

1. GLM Classification algorithm is used for building the classification model
2. Summary stats for the fitted model was printed using statsmodel. P-value for some variable was very high.

FEATURE SELECTION

1. Recursive feature elimination was used for same and final 15 features were kept for further analysis.
2. After checking the p-values now using statsmodel, they all were very small (<0.05)

VIF (VARIABLE INFLATION FACTOR)

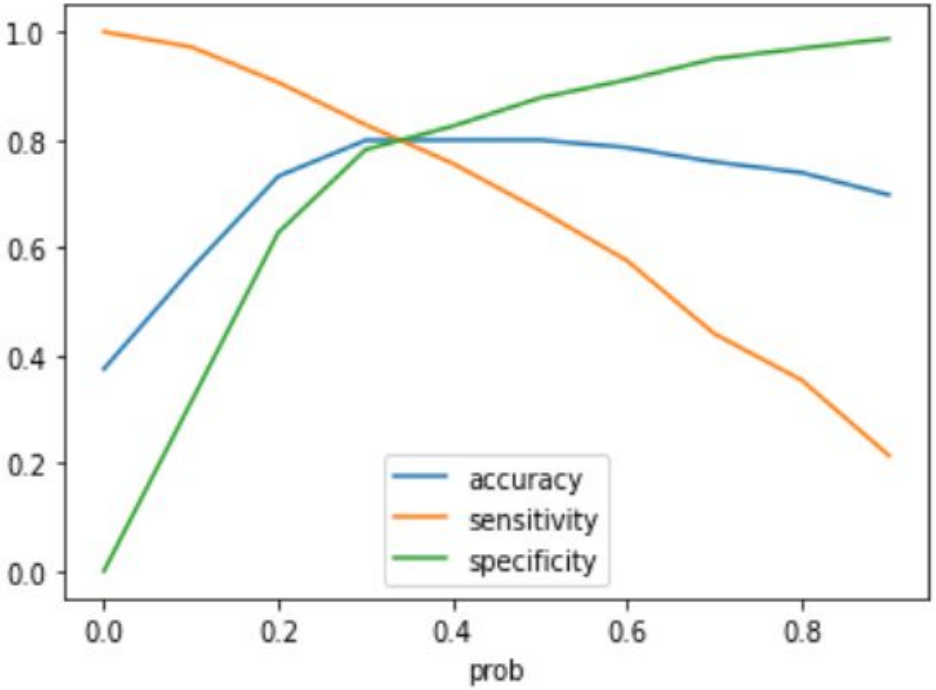
1. Now, checking the VIF for these 15 variables to check if there are any significant correlations left.
2. All the VIFs were very small (<5), thus the variables seem to be fine for the model

PREDICTION ON TRAINED DATASET

1. Taking the cutoff to be 0.5, and predicting the churn value for each lead, we calculated some performance metrics for it:Accuracy - 80%, sensitivity - 67% and specificity - 88%
2. Note: In our case, we want sensitivity to be good, because we don't want to miss any lead that can be our potential customer. Thus, we need to choose optimal cutoff point.

FINDING OPTIMAL CUTOFF

Plotting accuracy, specificity and sensitivity for all the points in list [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9] as cutoff values, we get:



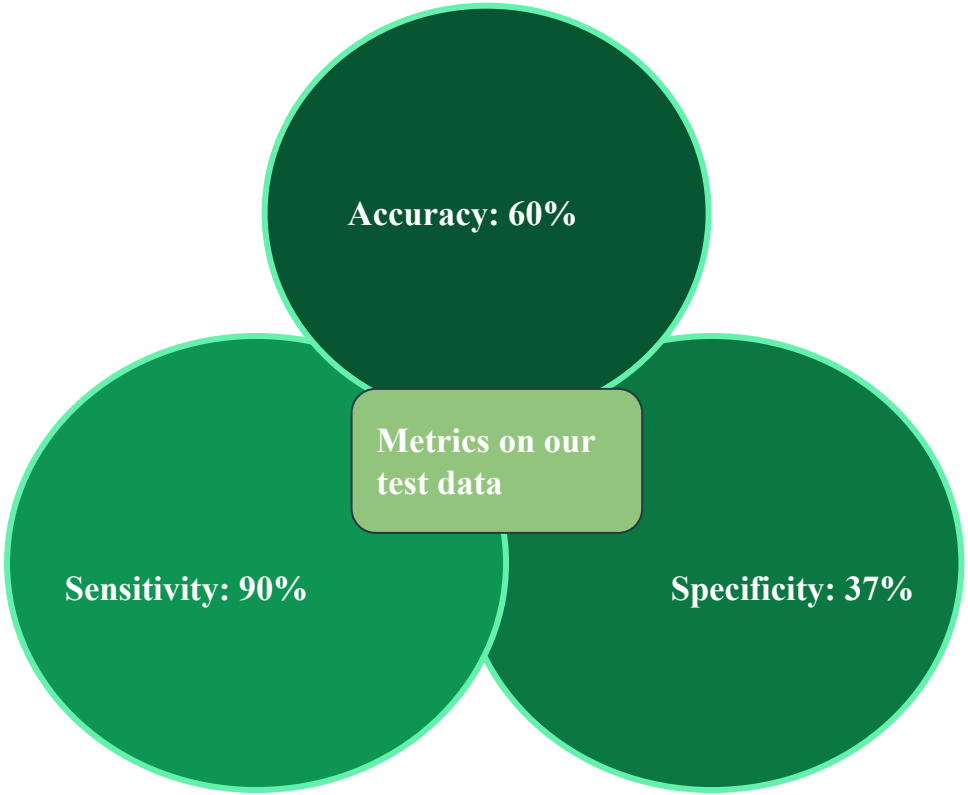
Here, we can see that they intersect at a value of ~0.35, here sensitivity, specificity and accuracy ~ 75%. Thus 0.35 seem to an optimal point

Finally using this cutoff, on predicting the churns on our train dataset, we get the following performance metrics:

1. Accuracy: 80%
2. Sensitivity: 80%
3. Specificity: 80%

Thus, this seems to be good model

PREDICTIONS ON TEST DATA



RESULTS

The features that we got after RFE are the most important metrics that the company can focus on for improving its conversion/churn rate.Finally with the predictions that we got on the training and testing dataset, we developed a dataframe that has a column Lead_Score that shows a number between 0 and 100 which indicates how likely is that lead going to churn. It is computed as follows:

LEAD_SCORE = CHURN PROBABILITY * 100

	Churn	Churn_Prob	final_predicted	Lead_Score
0	1	0.086270	0	8.626953
1	1	0.977273	1	97.727279
2	0	0.236533	0	23.653257
3	1	0.154779	0	15.477893
4	1	0.776051	1	77.605137