
Lead Scoring
Analysis
Badal Vishal

EXPLORATORY DATA ANALYSIS

- Data Cleaning & Treatment:
  - Dropped Lead Number and Prospect ID since they have all unique values
  - Converted 'Select' values to NaN
  - checking null values and percentage of null values in each row
  - dropping cols with more than 45% missing values

Categorical Attributes Analysis:

- checking value counts of Country column
- plotting spread of Country column
- Since India is the most common occurrence among the non-missing values replacing all missing values with India
- plotting spread of Country column after replacing NaN values
- creating a list of columns to be dropped
- checking value counts of "City" column
- plotting spread of City column after replacing NaN values
- checking value counts of Specialization column

- Lead may not have mentioned specialization because it was not in the list or maybe they are a students and don't have a specialization yet. So we will replace NaN values here with 'Not Specified’
- As per spread of Specialization column, combining Management Specializations because they show similar trends
- visualizing count of Variable based on Converted value
- In “What is your current occupation” imputing Nan values with mode "Unemployed“
- visualizing count of Variable based on Converted value
- Now, check value counts 'What matters most to you in choosing a course’
- replacing Nan values with Mode "Better Career Prospects“ & visualize
- checking value counts, this Column is worth Dropping. So appending to the cols_to_drop List

- Now, checking value counts of Tag variable and replacing Nan values with "Not Specified“
- visualizing count of Variable based on Converted value
- replacing tags with low frequency with "Other Tags“
- checking percentage of missing values for Tag
- Now, checking value counts of Lead Source column
- replacing Nan Values and combining low frequency values
- visualizing count of Variable based on Converted value

Inference

- Maximum number of leads are generated by Google and Direct traffic.
- Conversion Rate of reference leads and leads through welingak website is high.
- To improve overall lead conversion rate, focus should be on improving lead converion of olark chat, organic search,
- direct traffic, and google leads and generate more leads from reference and welingak website.

- Last Activity:
- replacing Nan Values and combining low frequency values
- Last Activity value counts
- Check the Null Values in All Columns
- Drop all rows which have Nan Values. Since the number of Dropped rows is less than 2%, it will not affect the model
- Checking percentage of Null Values in All Columns
- Lead Origin: visualizing count of Variable based on Converted value

Inference

- API and Landing Page Submission bring higher number of leads as well as conversion.
- Lead Add Form has a very high conversion rate but count of leads are not very high.
- Lead Import and Quick Add Form get very few leads.
- In order to improve overall lead conversion rate, we must improve lead conversion of API and Landing Page Submission origin and generate more leads from Lead Add Form.

- Do Not Email & Do Not Call :
- visualizing count of Variable based on Converted value
- checking value counts for Do Not Call, only 2 yes, add it to cols_to_drop
- IMBALANCED VARIABLES THAT CAN BE DROPPED
- adding imbalanced columns to the list of columns to be dropped as well
- Following columns : ['Search','Magazine','Newspaper Article','X Education Forums','Newspaper', 'Digital Advertisement','Through Recommendations', 'Receive More Updates About Our Courses', 'Update me on Supply Chain Content', 'Get updates on DM Content','I agree to pay the amount through cheque']
- checking value counts of last Notable Activity
- visualizing count of Variable based on Converted value
- checking value counts for variable last Notable Activity

- Dropping all columns added to cols_to_drop
- Check the % of Data that has Converted Values = 1: 38.0204328
- Checking correlations of numeric values : figure size, heatmap
- Total Visits : visualizing spread of variable with sns boxplot
- We can see presence of outliers here : checking percentile values for "Total Visits“
- Outlier Treatment: Remove top & bottom 1% of the Column Outlier values
- checking percentiles for "Total Time Spent on Website“
- visualizing spread of numeric variable
- Since there are no major Outliers for the above variable we don't do any Outlier Treatment for this above Column, Check for Page Views Per Visit: checking spread of "Page Views Per Visit"

- visualizing spread of numeric variable : 'Page Views Per Visit’ with sns boxplot
- Outlier Treatment: Remove top & bottom 1%
- checking Spread of "Total Visits" vs Converted variable
- Inference :
  - Median for converted and not converted leads are the close.
  - Nothng conclusive can be said on the basis of Total Visits.
- checking Spread of "Total Time Spent on Website" vs Converted variable
- Inference :
  - Leads spending more time on the website are more likely to be converted.
  - Website should be made more engaging to make leads spend more time.

- checking Spread of "Page Views Per Visit" vs Converted variable
- Inference
  - Median for converted and unconverted leads is the same.
  - Nothing can be said specifically for lead conversion from Page Views Per Visit
- checking missing values in leftover columns
- There are no missing values in the columns to be analyzed further

- Dummy Variable Creation:
  - getting a list of categorical columns
  - List of variables to map, and Defining the map function
  - def binary_map(x):
  - return x.map({'Yes': 1, "No": 0})
  - Applying the function to the housing list
  - getting dummies and dropping the first column and adding the results to the master dataframe
  - dropping the original columns after dummy variable creation
  - Train-Test Split & Logistic Regression Model Building
  - Splitting the data into train and test
  - scaling numeric columns

- Model Building using Stats Model & RFE:
  - running RFE with 15 variables as output
  - list of RFE supported columns
  - BUILDING MODEL #1
    - X_train_sm = sm.add_constant(X_train[col])
    - logm1 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
    - res = logm1.fit()
    - res.summary()
    - p-value of variable Lead Source_Referral Sites is high, so we can drop it.
  - BUILDING MODEL #2
    - Check for the VIF values of the feature variables
    - Create a dataframe that will contain the names of all the feature variables and their respective VIFs
    - There is a high correlation between two variables so we drop the variable with the higher valued VIF value
    - #dropping variable with high VIF 'Last Notable Activity_SMS Sent'

- BUILDING MODEL #3
  - X_train_sm = sm.add_constant(X_train[col])
  - logm3 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
  - res = logm3.fit()
  - res.summary()
  - Create a dataframe that will contain the names of all the feature variables and their respective VIFs
  - So the Values all seem to be in order so now, Moving on to derive the Probabilities, Lead Score, Predictions on Train Data:
  - Getting the Predicted values on the train set
  - from sklearn import metrics
  - # Confusion matrix
  - confusion = metrics.confusion_matrix(y_train_pred_final.Converted, y_train_pred_final.Predicted)
  - Let's check the overall accuracy : 0.9250
  - Let's see the sensitivity of our logistic regression model : 0.8821
  - Let us calculate specificity : 0.9513
  - Calculate False Postive Rate - predicting conversion when customer does not have convert: 0.048
  - positive predictive value : 0.9175
  - Negative predictive value :0.9292
  - PLOTTING ROC CURVE : area 0.97

- The ROC Curve should be a value close to 1. We are getting a good value of 0.97 indicating a good predictive model.

- Finding Optimal Cutoff Point
- Above we had chosen an arbitrary cut-off value of 0.5. We need to determine the best cut-off value and
- the below section deals with that:
  - Let's create columns with different probability cutoffs
  - Now let's calculate accuracy sensitivity and specificity for various probability cutoffs.
  - # TP = confusion[1,1] # true positive
  - # TN = confusion[0,0] # true negatives
  - # FP = confusion[0,1] # false positives
  - # FN = confusion[1,0] # false négatives
  - Let's plot accuracy sensitivity and specificity for various probabilities
  - From the curve, 0.3 is the optimum point to take it as a cutoff probability

- Let's check the overall accuracy : 0.9229
- Let's see the sensitivity of our logistic regression model : 0.91698
- Let us calculate specificity : 0.9265
- Calculate False Postive Rate - predicting conversion when customer does not have convert : 0.0734
- Positive predictive value : 0.8847
- Negative predictive value : 0.9478
- Looking at the confusion matrix again

- Observation:
- So as we can see above the model seems to be performing well. The ROC curve has a value of 0.97, which is very good.
- We have the following values for the Train Data:
- Accuracy : 92.29%
- Sensitivity : 91.70%
- Specificity : 92.66%
- Some of the other Stats are derived below, indicating the False Positive Rate, Positive Predictive Value,
- Negative Predictive Values, Precision & Recall
- Precision : 0.8847
- Recall : 0.9169

- scaling test set
- PREDICTIONS ON TEST SET
- Converting y_pred to a dataframe which is an array
- Converting y_test to dataframe
- Putting CustID to index
- Removing index for both dataframes to append them side by side
- Appending y_test_df and y_pred_1
- Renaming the column 'Converted_prob’ and Rearranging the columns
- Let's check the overall accuracy : 0.9277
- Let's see the sensitivity of our logistic regression model : 0.9198
- Let us calculate specificity : 0.9325

Observation

- # After running the model on the Test Data these are the figures we obtain:
- # Accuracy : 92.78%
- # Sensitivity : 91.98%
- # Specificity : 93.26%
- # Final Observation:
- Train Data:
- # Accuracy : 92.29%
- # Sensitivity : 91.70%
- # Specificity : 92.66%
- Test Data:
- # Accuracy : 92.78%
- # Sensitivity : 91.98%
- # Specificity : 93.26%
- # The Model seems to predict the Conversion Rate very well and we should be able to give the CEO confidence in making good calls based on this model

Thank you