Lead Scoring Dataset

Name: Sumit Bansod Roll no :220940325081

Steps followed

1. Importing Data, Inspecting the Dataframe

2. Data Preparation (Encoding Categorical Variables, Handling Null Values)

3. EDA (univariate analysis, outlier detection, checking data imbalance)

4. Dummy Variable Creation

5. Test-Train Split

6. Feature Scaling

7. Looking at Correlations

8. Model Building (Feature Selection Using RFE, Improvising the model further inspecting adjusted R-squared, VIF and p-values)

9. Build final model

Detailed Solution (Implementation done in Python):

- We are using Leads.csv dataset for this case study.
- Regression Model and evaluating final model’s performance.
- Looking at the dataset we observed that following things need to be done:
- we need to encode the categorical columns into numeric values in order to use them in Logistic Regression
- Missing value handling is required for some of the features.
- Dropping columns having more than 70% null values

Data Preparation : Missing Value Handling

- There are some categorical features having a label as “SELECT”. This is the same as missing values. So converting SELECT into the NaN
- After identifying all the missing data, dropped columns having more than 45% null values

Categorical Column Analysis

- There are too many variations in the columns (‘Asymmetrique Activity Index’,’Asymmetrique Activity Score’,’Asymmetrique Profile Index’,’Asymmetrique Profile Score’) and it is not safer to impute any values in the columns and hence we will drop these columns with very high percentage of missing data
- We can impute the MUMBAI into all the NULLs as most of the values belong to MUMBAI
- Since there is no significant difference among top 3 specialization , hence it will be safer to impute NaN with Not Specified

- For Tags column, more than 30% data is for “Will revert after reading the email” and hence we can impute NULLS with Will revert after reading the email

- For “what is your current occupation” column, we can impute NULLS with “unemployed”

- More than 99% data is of “Better Career Prospects” and hence it is safer to impute NULLS with this value

- More than 85% data is of “Unemployed” and hence it is safer to impute NULLS with this value

- More than 95% data is of “India” and hence it is safer to impute NULLS with this value

- 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 conversion of olark chat, organic search, direct traffic, and google leads and generate more leads from reference and welingak website.

- Drop all rows which have Nan Values. Since the number of Dropped rows is less than 2%, it will not affect the model

Visualization

- Visualizing count of Variable based on Converted value using countplot

- Using countplot we observe that API and Landing Page Submission bring a 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 have to improve lead conversion of API and Landing Page Submission origin and generate more leads from Lead Add Form.

  - Websites can be made more appealing so as to increase the time of the Users on websites

  -

  - Converting all the low count categories to the 'Others' category and plotting again.

  - The count of last activity as "Email Opened" is max

  - The conversion rate of SMS sent as last activity is maximum

Numeric Feature Analysis

- Here 'Converted' is the response column, so we'll do our analysis taking response column into consideration

- Check the correlation between all the numeric column using heatmap

- For column "TotalVisits" and "Page Views Per Visit" remove top and bottom 1% outliers

- By using all this process our data is clean now and ready to process

- Convert all the column which have categorical data into numerical data by using dummy variable

Creating Dummy Variable

Now After data cleaning and analysis our data has no missing value.

So , now we creating dummy variables for column ['Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity','Specialization', 'What is your current occupation', 'Tags', 'City','A free copy of Mastering The Interview', 'Last Notable Activity'].

.

Splitting Data into Training and Test set

- Next, the dataset was split into training and test sets, to train the model first with a chunk of data and then evaluate its performance on unseen data.

- Feature Scaling

- Feature Scaling is required before Logistic Regression to bring all the features in the same scale, this ensures that features with high magnitude are not given higher importance by the Logistic Regression Model.

Scaling of Data

Some of our columns have larger values which can create biasing in the model so for avoiding that we need to do scaling.

Model Building

```
from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()
```

After printing the summary we can see that the p-value of variable Lead Source_Referral Sites is high, so we can drop it.

After doing that on again printing the summary again we can see that Since 'All' the p-values are less we can check the Variance Inflation Factor to see if there is any correlation between the variables.

Now we will check for correlation between columns by using VIF so we can see that There is a high correlation between two variables so we drop the variable with the higher valued VIF value.

So dropping columns with high VIF Last Notable Activity_SMS Sent.

On printing summary we can see that all the Values all seem to be in order so now, Moving on to derive the Probabilities, Lead Score, Predictions on Train Data

So now our model is ready to do prediction.

Model has an accuracy of : 89.08%