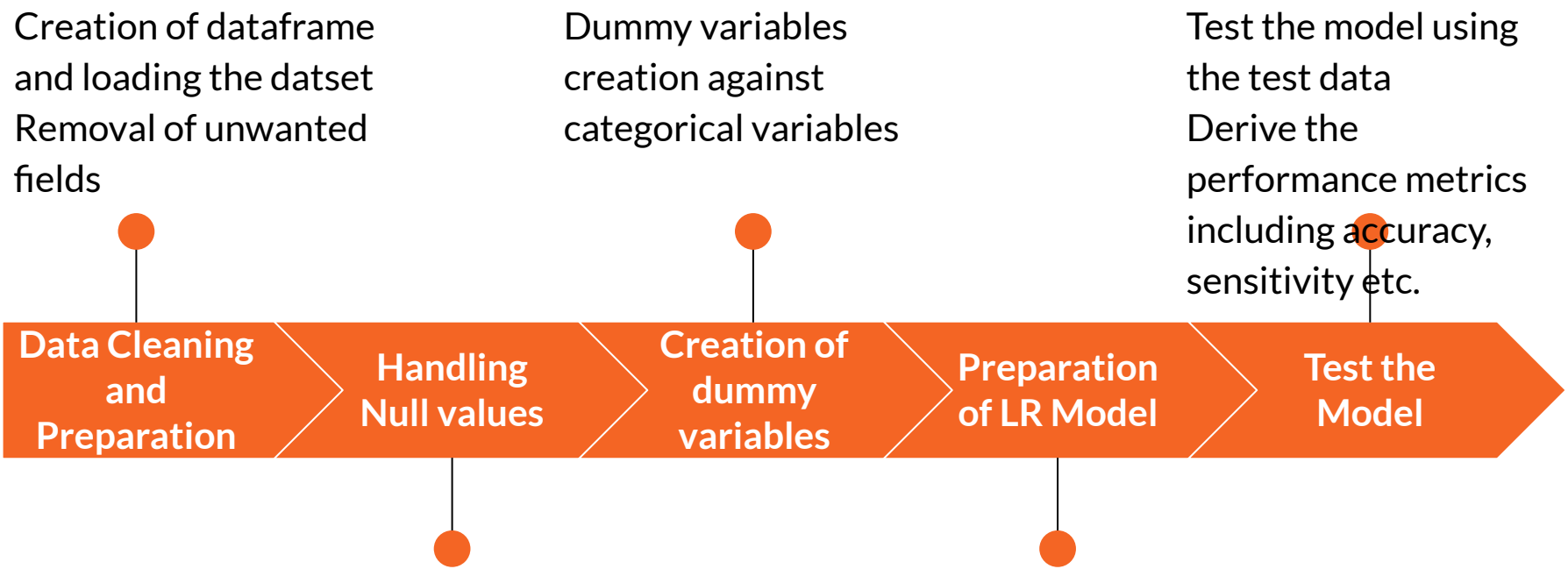
Case Study: Lead Scoring

Tulika Joseph | Johny Mathew | 18.06.2024

Problem Statement

Build a logistic regression model to assign a lead score between 0 and 100 to each of the leads which can be used by the company to target potential leads


Creation of dataframe and loading the datset
Removal of unwanted fields
Dummy variables creation against categorical variables
Test the model using the test data
Derive the performance metrics including accuracy, sensitivity etc.
Data Cleaning and Preparation
Handling Null values
Creation of dummy variables
Preparation of LR Model
Test the Model
Removal of columns with large number of null fields
Remove fields with repeating data
Split into test and train data
Scale numeric fields
Preparation of Logistic Regression model

Approach

1. Data Cleaning and Preparation

- Import various functions as needed
- Load the dataframe
- Check data quality, info, shape
- Identify null values in each field

2. Handling Null Values

- Removal of fields with > 3000 null values
- Remove fields without impact on target variable (City, State)
- Fields with large number of values as ‘Select’
- 31% of rows removed after the process
- Fields such as prospect id, lead number dropped

Fields Dropped: 'Do Not Call', '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'

3. Creation of Dummy variables

- Dummy variables created for categorical variables
- Corresponding variables dropped
- Specialization field with large number of blanks ‘Select’ dropped instead of drop_first

4. Test/Train Split and scaling

- MinMaxScaler used
- 70:30 split on data for Train:Test

Fields Dropped: 'Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity', 'Specialization', 'What is your current occupation', 'A free copy of Mastering The Interview', 'Last Notable Activity'

4. Building the Logistic Regression Model

- Regression model built with max 1000 iterations
- Fields identified when run for 15 variables include

```
In [53]: #Storing the selected variables identified by RFE
         cols = X_train.columns[rfe.support_]
         cols
```

```
Out[53]: Index(['TotalVisits', 'Total Time Spent on Website',
                'Lead Origin_Lead Add Form', 'Lead Source_Olark Chat',
                'Lead Source_Reference', 'Lead Source_Welingak Website',
                'Do Not Email_Yes', 'Last Activity_Converted to Lead',
                'Last Activity_Olark Chat Conversation', 'Last Activity_SMS Sent',
                'What is your current occupation_Student',
                'What is your current occupation_Unemployed',
                'What is your current occupation_Working Professional',
                'Last Notable Activity_Had a Phone Conversation',
                'Last Notable Activity_Unreachable'],
               dtype='object')
```

4. Building the Logistic Regression Model

- Checking for the P values and high VIF removed further fields, bringing the model down to 13 variables

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.6474	0.585	-1.107	0.268	-1.793	0.498
TotalVisits	4.0447	1.199	3.375	0.001	1.696	6.394
Total Time Spent on Website	4.3198	0.184	23.421	0.000	3.958	4.681
Lead Origin_Lead Add Form	3.5342	0.227	15.553	0.000	3.089	3.980
Lead Source_Olark Chat	1.5566	0.126	12.366	0.000	1.310	1.803
Lead Source_Welingak Website	2.0778	0.752	2.764	0.006	0.604	3.551
Do Not Email_Yes	-1.5573	0.193	-8.079	0.000	-1.935	-1.179
Last Activity_Converted to Lead	-1.1403	0.238	-4.795	0.000	-1.606	-0.674
Last Activity_Olark Chat Conversation	-1.3210	0.184	-7.163	0.000	-1.682	-0.960
Last Activity_SMS Sent	1.0674	0.084	12.740	0.000	0.903	1.232
What is your current occupation_Student	-1.3919	0.617	-2.255	0.024	-2.602	-0.182
What is your current occupation_Unemployed	-1.4870	0.581	-2.559	0.010	-2.626	-0.348
What is your current occupation_Working Professional	1.3025	0.613	2.125	0.034	0.101	2.504
Last Notable Activity_Unreachable	2.5712	0.814	3.158	0.002	0.975	4.167

5. Model Evaluation

- Model run on train set
- Probability values stored as an array
- Arbitrarily selecting cut off as 0.5 accuracy of model was tested
- Optimal cutoff identified by using ROC curve and plotting against various cutoffs

```
# Let's check the overall accuracy.
metrics.accuracy_score(y_train_pred_final.Converted,
```

```
0.7821116341627438
```

```
# Let's evaluate the other metrics as well

TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
```

```
# Calculate the sensitivity

TP/(TP+FN)
```

```
0.7331144465290806
```

```
# Calculate the specificity

TN/(TN+FP)
```

```
0.8269643623872907
```

5. Model Evaluation

0.42 identified as optimal cutoff basis the plot against accuracy, sensitivity and specificity for various cut off points

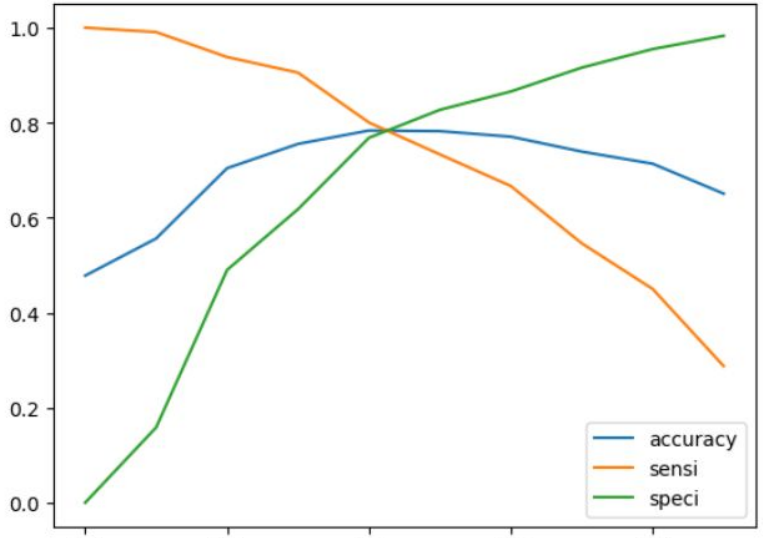
Final predictor variable adjusted to 0.42 as cut off value

Accuracy and sensitivity further calculated again.

	prob	accuracy	sensi	speci
0.0	0.0	0.477920	1.000000	0.000000
0.1	0.1	0.556153	0.990619	0.158437
0.2	0.2	0.704102	0.938086	0.489910
0.3	0.3	0.755212	0.905253	0.617862
0.4	0.4	0.783457	0.800188	0.768141
0.5	0.5	0.782112	0.733114	0.826964
0.6	0.6	0.770455	0.666510	0.865608
0.7	0.7	0.739072	0.545966	0.915844
0.8	0.8	0.713517	0.449812	0.954916
0.9	0.9	0.650527	0.287523	0.982825

```
# Let's plot it as well

cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



6. Making Predictions on the Test set

Test set scaled using original scaler on numeric fields

Dropped fields from previous models dropped again

Model run and target variable plotted using the same cutoff of 0.42

In [111]:
```
# Let's check the overall accuracy

metrics.accuracy_score(y_pred_final['Converted'], y_pred_final.final_predicted
```

Out[111]:
```
0.7986401673640168
```

In [112]:
```
confusion2 = metrics.confusion_matrix(y_pred_final['Converted'],
                                      y_pred_final.final_predicted)
confusion2
```

Out[112]:
```
array([[788, 191],
       [194, 739]], dtype=int64)
```

In [113]:
```
TP = confusion2[1, 1]  # true positive
TN = confusion2[0, 0]  # true negatives
FP = confusion2[0, 1]  # false positives
FN = confusion2[1, 0]  # false negatives
```

In [114]:
```
# Calculate sensitivity
TP / float(TP+FN)
```

Out[114]:
```
0.7920685959271169
```

In [115]:
```
# Calculate specificity
TN / float(TN+FP)
```

Out[115]:
```
0.804902962206333
```

Findings and summaries

1. Top variables identified basis the model built as
- Total Visits
- Total Time Spent on Website
- Lead Origin_Lead Add form

2. Top Dummy variables identified as
- Lead Origin Lead Add Form
- Last Notable Activity Unreachable
- Lead Source_Welingak Website

3. Optimizing for high sales during peak season

This is done by optimizing for high Recall, minimizing false negatives, thus making sure that a healthy catch of leads are shared with the calling teams to engage with

4. When the company has achieved targets, only very high confidence leads are to be pushed and others are to be engaged with via other cost effective channels such as e-mails, CRM etc. By setting a much higher cutoff we can ensure that only the most critical leads are passed on to the system.

Thank You!

Tulika Joseph | Johny Mathew | 18.06.2024