Lead Score for X education system

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.
- The company markets its courses on several websites and search engines like Google. Once these people land on the website, they might browse the courses or fill up a form for the course or watch some videos. When these people fill up a form providing their email address or phone number, they are classified to be a lead. Moreover, the company also gets leads through past referrals. Once these leads are acquired, employees from the sales team start making calls, writing emails, etc. Through this process, some of the leads get converted while most do not. The typical lead conversion rate at X education is around 30%.
- Now, although X Education gets a lot of leads, its lead conversion rate is very poor. For example, if, say, they acquire 100 leads in a day, only about 30 of them are converted. 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.
- 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 higher lead score have a higher conversion chance and the customers with 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%.

Approach to solve the Business Problem

1. Importing Data
2. Inspecting the Dataframe
3. Data Preparation (Encoding Categorical Variables, Handling Null Values)
4. EDA (univariate analysis, outlier detection, checking data imbalance)
5. Dummy Variable Creation
6. Test-Train Split
7. Feature Scaling
8. Looking at Correlations
9. Model Building (Feature Selection Using RFE, Improvising the model further inspecting adjusted R-squared, VIF and p-vales)
10. Build final model
11. Model evaluation with different metrics Sensitivity, Specificity

Importing the data

[394]:
```
import pandas as pd,numpy as np
import matplotlib.pyplot as plt,seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

Loading the data

[395]:
```
df = pd.read_csv('Leads.csv')
df.head()
```

[395]:

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	...	Get updates on DM Content	Lead Profile	City	Asymmetrique Activity Index	Asymmetrique Profile Index	Asymmetrique Activity Scor
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	...	No	Select	Select	02.Medium	02.Medium	15.(
1	2a272436-5132-4136-86fa-	660728	API	Organic Search	No	No	0	5.0	674	2.5	...	No	Select	Select	02.Medium	02.Medium	15.(

Inspecting the dataframe

Step 2:Inspecting the dataset

```
99]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9240 entries, 0 to 9239
Data columns (total 37 columns):
 #   Column                                          Non-Null Count  Dtype
---  ------                                          --------------  -----
 0   Prospect ID                                     9240 non-null   object
 1   Lead Number                                     9240 non-null   int64
 2   Lead Origin                                     9240 non-null   object
 3   Lead Source                                     9204 non-null   object
 4   Do Not Email                                    9240 non-null   object
 5   Do Not Call                                     9240 non-null   object
 6   Converted                                       9240 non-null   int64
 7   TotalVisits                                     9103 non-null   float64
 8   Total Time Spent on Website                     9240 non-null   int64
 9   Page Views Per Visit                            9103 non-null   float64
 10  Last Activity                                   9137 non-null   object
 11  Country                                         6779 non-null   object
 12  Specialization                                  7802 non-null   object
 13  How did you hear about X Education              7033 non-null   object
 14  What is your current occupation                 6550 non-null   object
 15  What matters most to you in choosing a course   6531 non-null   object
 16  Search                                          9240 non-null   object
 17  Magazine                                        9240 non-null   object
```

```
df.describe()
```

	Lead Number	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Asymmetrique Activity Score	Asymmetrique Profile Score
count	9240.000000	9240.000000	9103.000000	9240.000000	9103.000000	5022.000000	5022.000000
mean	617188.435606	0.385390	3.445238	487.698268	2.362820	14.306252	16.344883
std	23405.995698	0.486714	4.854853	548.021466	2.161418	1.386694	1.811395
min	579533.000000	0.000000	0.000000	0.000000	0.000000	7.000000	11.000000
25%	596484.500000	0.000000	1.000000	12.000000	1.000000	14.000000	15.000000
50%	615479.000000	0.000000	3.000000	248.000000	2.000000	14.000000	16.000000
75%	637387.250000	1.000000	5.000000	936.000000	3.000000	15.000000	18.000000
max	660737.000000	1.000000	251.000000	2272.000000	55.000000	18.000000	20.000000

DataProcessing

```
varlist =  ['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','A free copy of Mastering The Interview',
         'Do Not Email','Do Not Call']

def binary_variable(x):
    return x.map({'Yes':1,'No':0})
df[varlist] = df[varlist].apply(binary_variable)
df.head()
```

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	...	Get updates on DM Content	Lead Profile	City	Asymmetrique Activity Index	Asymmetrique Profile Index	Asymmetriq Activity Sc
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	0	0	0	0.0	0	0.0	...	0	Select	Select	02.Medium	02.Medium	1
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	0	0	0	5.0	674	2.5	...	0	Select	Select	02.Medium	02.Medium	1
	8cc8c611-		Landing														

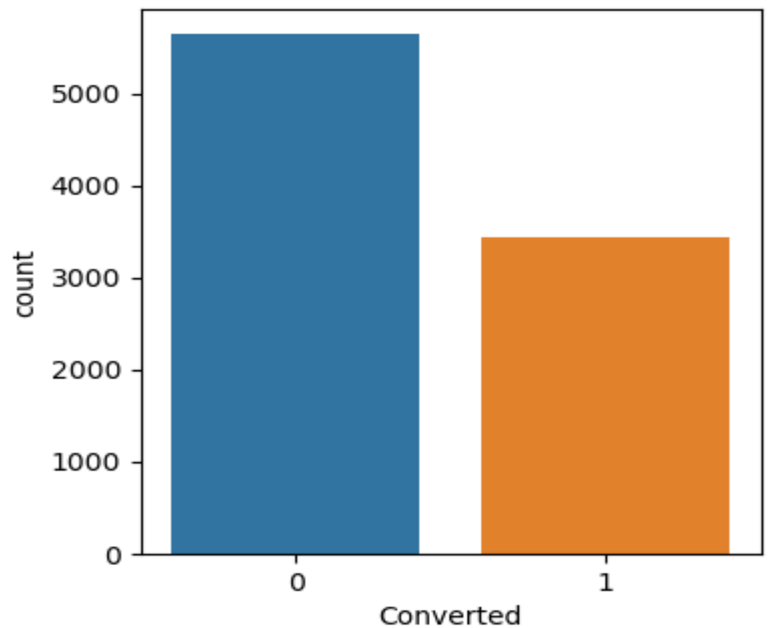
```python
## Imputing the null values
df['City']=df['City'].replace(np.nan,'Mumbai')
df['Country']= df['Country'].replace(np.nan,'India')
df['Specialization']= df['Specialization'].replace(np.nan,'Other_Specialization')
df['What is your current occupation']=df['What is your current occupation'].replace(np.nan,'Unemployed')
df['What matters most to you in choosing a course']=df['What matters most to you in choosing a course'].replace(np.nan,'Better Career Prospects')
```

```python
df['Lead Quality']= df['Lead Quality'].replace(np.nan,'Not Sure')
df['Tags'] = df['Tags'].replace(np.nan,'Will revert after reading the email')
```

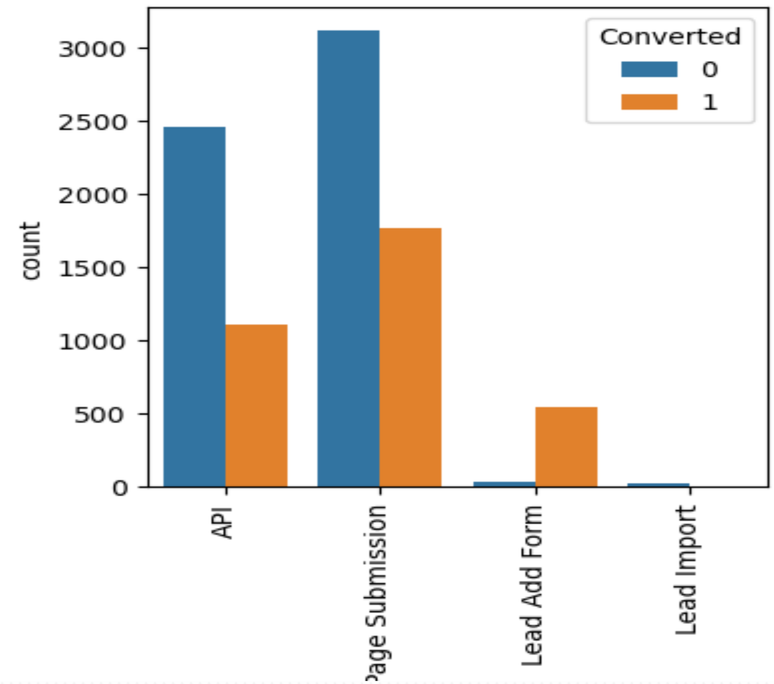
```python
## for 2% or less than missing value we will treat it like it
df.dropna(inplace=True)
```

EDA Analysis of Data Set

```
plt.figure(figsize=(4,4))
sns.countplot(x='Converted',data=df)
plt.show()
```




```
sns.countplot(x='Lead Origin',hue='Converted',data=df)
plt.xticks(rotation=90)
plt.show()
```



Creating Dummy variable

```
dummy=pd.get_dummies(df[['Lead Origin', 'Lead Source','Last Activity',
       'Specialization', 'What is your current occupation', 'Tags',
       'Lead Quality', 'Last Notable Activity','City']],drop_first=True).astype(int)
```

```
dummy.head()
```

	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	Lead Source_Organic Search	Lead Source_Other Website	Lead Source_Reference	Lead Source_Referral Sites	...
0	0	0	0	0	0	1	0	0	0	0	...
1	0	0	0	0	0	0	1	0	0	0	...
2	1	0	0	0	0	0	0	0	0	0	...
3	1	0	0	0	0	0	0	0	0	0	...
4	1	0	0	0	1	0	0	0	0	0	...

5 rows × 80 columns

Splitting the Data into Train and test set

```python
from sklearn.model_selection import train_test_split
```

```python
X=df.drop(['Prospect ID','Converted'],axis=1)
X.head()
```

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	Lead Source_Organic Search	...
0	0.0	0	0.0	0	0	0	0	0	1	0	...
1	5.0	674	2.5	0	0	0	0	0	0	1	...
2	2.0	1532	2.0	1	0	0	0	0	0	0	...
3	1.0	305	1.0	1	0	0	0	0	0	0	...
4	2.0	1428	1.0	1	0	0	0	1	0	0	...

```python
y=df['Converted']
y.head()
```

```
0    0
1    0
2    1
3    0
4    1
Name: Converted, dtype: int64
```

```python
## Spliting the data into train and test dataset
X_train,X_test,y_train,y_test = train_test_split(X,y,train_size=0.8,test_size=0.2,random_state=100)
```

Feature Scaling of the Data

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()

X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]=scaler.fit_transform(X_train[['TotalVisits','Total Time Spent on Website'

X_train.head()
```

	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Lead Origin_Landing Page Submission	Lead Origin_Lead Add Form	Lead Origin_Lead Import	Lead Source_Facebook	Lead Source_Google	Lead Source_Olark Chat	Lead Source_Organic Search	...	Last Notable Activity_Resubscribed to emails	Last N Activi
160	-0.071614	0.961655	0.298374	1	0	0	0	0	0	0	...	0	
2267	-1.147903	-0.886605	-1.125450	0	0	0	0	0	1	0	...	0	
8895	-1.147903	-0.886605	-1.125450	0	0	0	0	0	1	0	...	0	
854	0.287149	2.136489	0.772982	1	0	0	0	0	0	1	...	0	
3640	0.287149	-0.505974	0.772982	1	0	0	0	1	0	0	...	0	

Analyzing the Correlations

```
data_corr = df.drop('Prospect ID',axis=1)
```

```
conv_corr = data_corr.corr()
conv_corr_unstacked = conv_corr.unstack().sort_values()
conv_corr.where(np.triu(np.ones(conv_corr.shape), k=1).astype(bool)).stack().sort_values(ascending=False).head(10)
```

```
Lead Origin_Lead Import               Lead Source_Facebook                        0.983684
Last Activity_Unsubscribed            Last Notable Activity_Unsubscribed          0.872656
Lead Origin_Lead Add Form             Lead Source_Reference                       0.866191
Last Activity_Email Opened            Last Notable Activity_Email Opened          0.861636
Last Activity_SMS Sent                Last Notable Activity_SMS Sent              0.853102
Last Activity_Email Link Clicked      Last Notable Activity_Email Link Clicked    0.800686
TotalVisits                           Page Views Per Visit                        0.737996
Last Activity_Page Visited on Website Last Notable Activity_Page Visited on Website 0.691811
Last Activity_Unreachable             Last Notable Activity_Unreachable           0.594369
Last Activity_Other Activity          Last Notable Activity_Had a Phone Conversation 0.576457
dtype: float64
```

```
# Dropping highly correlated features

X_test = X_test.drop(['Lead Source_Facebook','Last Notable Activity_Unsubscribed','Last Notable Activity_SMS Sent',
                      'Last Notable Activity_Email Opened','Last Notable Activity_Unreachable','Last Notable Activity_Email Link Clicked','Last Notable
X_train = X_train.drop(['Lead Source_Facebook','Last Notable Activity_Unsubscribed','Last Notable Activity_SMS Sent',
                      'Last Notable Activity_Email Opened','Last Notable Activity_Unreachable','Last Notable Activity_Email Link Clicked','Last Notable
```

Model Building

```python
import statsmodels.api as sm
```

```python
# Logistic Regression Model
logm1 = sm.GLM(y_train,(sm.add_constant(X_train)),family=sm.families.Binomial())
logm1.fit().summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	**No. Observations:**	7259
Model:	GLM	**Df Residuals:**	7182
Model Family:	Binomial	**Df Model:**	76
Link Function:	Logit	**Scale:**	1.0000
Method:	IRLS	**Log-Likelihood:**	-1494.9
Date:	Mon, 29 Apr 2024	**Deviance:**	2989.8
Time:	22:14:19	**Pearson chi2:**	5.43e+04
No. Iterations:	23	**Pseudo R-squ. (CS):**	0.6003
Covariance Type:	nonrobust		

Feature Selection Using RFE

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
```

```
from sklearn.feature_selection import RFE
rfe = RFE(estimator=logreg, n_features_to_select=13)
rfe = rfe.fit(X_train, y_train)
```

```
rfe.support_
```

```
array([False, False, False, False, False, False, False, False, False,
       False, False, False,  True, False, False, False, False, False,
       False, False,  True, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False, False, False, False, False, False, False, False, False,
       False,  True,  True, False, False, False,  True, False, False,
        True,  True,  True,  True, False, False,  True,  True, False,
       False, False, False, False,  True,  True, False, False, False,
       False, False, False, False])
```

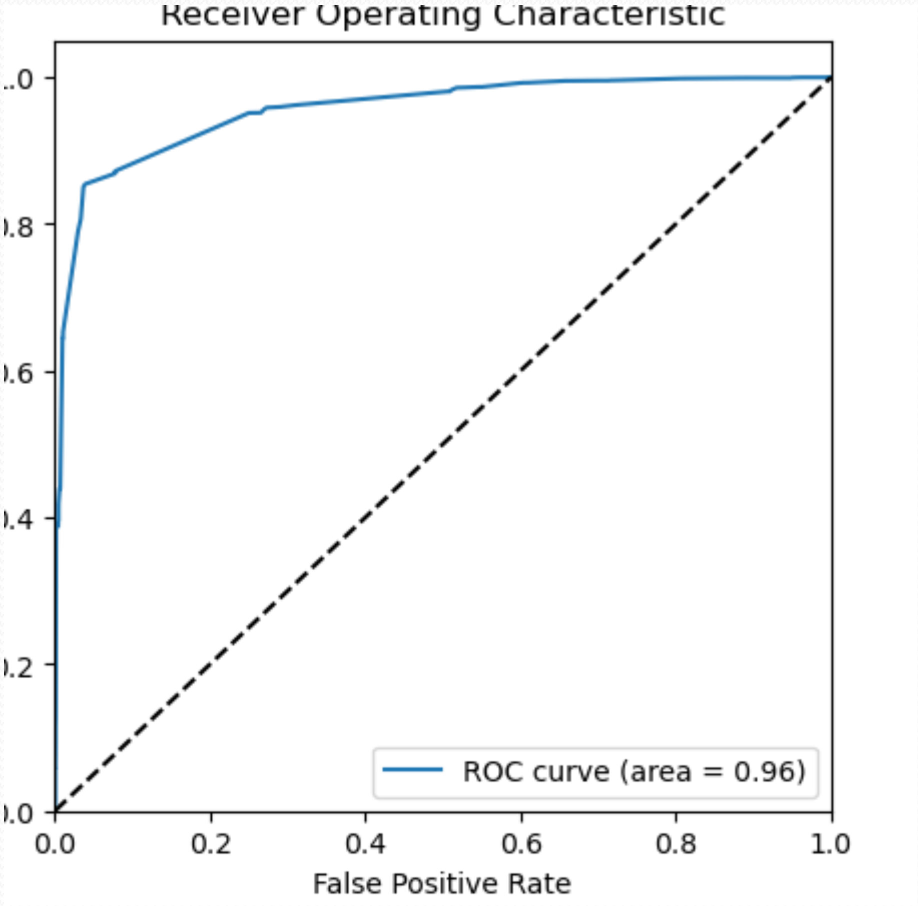
```
list(zip(X_train.columns,rfe.support_,rfe.ranking_))
```

```
[('TotalVisits', False, 47),
 ('Total Time Spent on Website', False, 4),
 ('Page Views Per Visit', False, 46),
 ('Lead Origin_Landing Page Submission', False, 14),
 ('Lead Origin_Lead Add Form', False, 2),
 ('Lead Origin_Lead Import', False, 20),
```

Creating ROC Curve

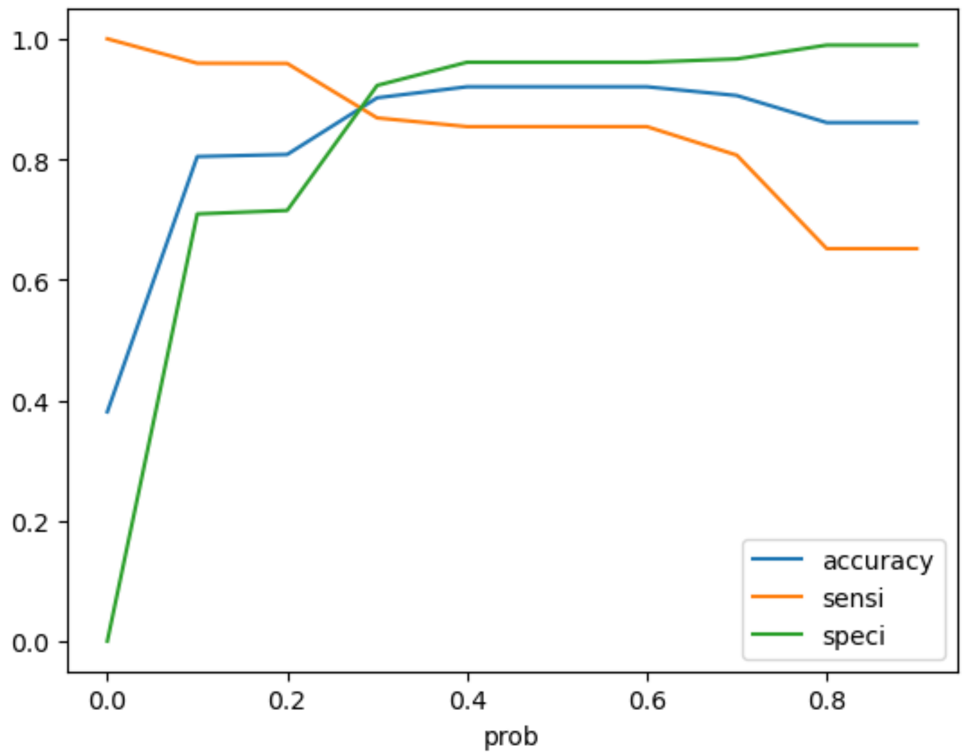
- An ROC curve demonstrates several things:- It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
- The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the test.
- The closer the curve comes to the 45-degree diagonal of the ROC space, the less accurate the test.

ROC Curve


Receiver Operating Characteristic
.0
.8
.6
.4
.2
.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate
ROC curve (area = 0.96)

Finding the optimal cut-off point

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



From the curve above, 0.3 is the optimum point to take it as a cutoff probability.

Making prediction on final test set

```
X_test[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']]=scaler.transform(X_test[['TotalVisits','Total Time Spent on Website
```

```
X_test = X_test[col]
X_test.head()
```

	Lead Source_Welingak Website	Last Activity_SMS Sent	Tags_Busy	Tags_Closed by Horizzon	Tags_Lost to EINS	Tags_Ringing	Tags_Will revert after reading the email	Tags_switched off	Lead Quality_Not Sure	Lead Quality_Worst	Last Notable Activity_Modified	Last No Activity_ Convers
3271	0	0	0	0	0	0	1	0	1	0	0	
1490	0	0	0	0	0	0	1	0	0	0	0	
7936	0	0	0	0	0	0	1	0	1	0	0	
4216	0	0	0	1	0	0	0	0	0	0	1	
3830	0	0	0	0	0	0	1	0	1	0	0	

```
X_test_sm = sm.add_constant(X_test)
```

Conclusion

- The logistic regression model predicts the probability of the target variable having a certain value, rather than predicting the value of the target variable directly. Then a cutoff of the probability is used to obtain the predicted value of the target variable.
- Here, the logistic regression model is used to predict the probabilty of conversion of a customer.
- Optimum cut off is chosen to be 0.27 i.e. any lead with greater than 0.27 probability of converting is predicted as Hot Lead (customer will convert) and any lead with 0.27 or less probability of converting is predicted as Cold Lead (customer will not convert)
- Our final Logistic Regression Model is built with 14 features.
- Features used in final model are ['Do Not Email', 'Lead Origin_Lead Add Form', 'Lead Source_Welingak Website', 'Last Activity_SMS Sent', 'Tags_Busy', 'Tags_Closed by Horizzon', 'Tags_Lost to EINS', 'Tags_Ringing', 'Tags_Will revert after reading the email', 'Tags_switched off', 'Lead Quality_Not Sure', 'Lead Quality_Worst', 'Last Notable Activity_Modified', 'Last Notable Activity_Olark Chat Conversation']
- The top three categorical/dummy variables in the final model are ‘Tags_Lost to EINS’, ‘Tags_Closed by Horizzon’, ‘Lead Quality_Worst’ with respect to the absolute value of their coefficient factors.