

# Lead Scoring Assignment

**Jameel Amer**

# Index

Business Understanding & Problem Statement

Problem Solving Methodology

Data Cleaning & Preparation

Build the Module

Predictions on test set

Feature Importance

Recommendations

# Business Understanding & Problem Statement

- ▶ Business Understanding
- ▶ Problem Statement

# Business Understanding


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.

# Problem Statement

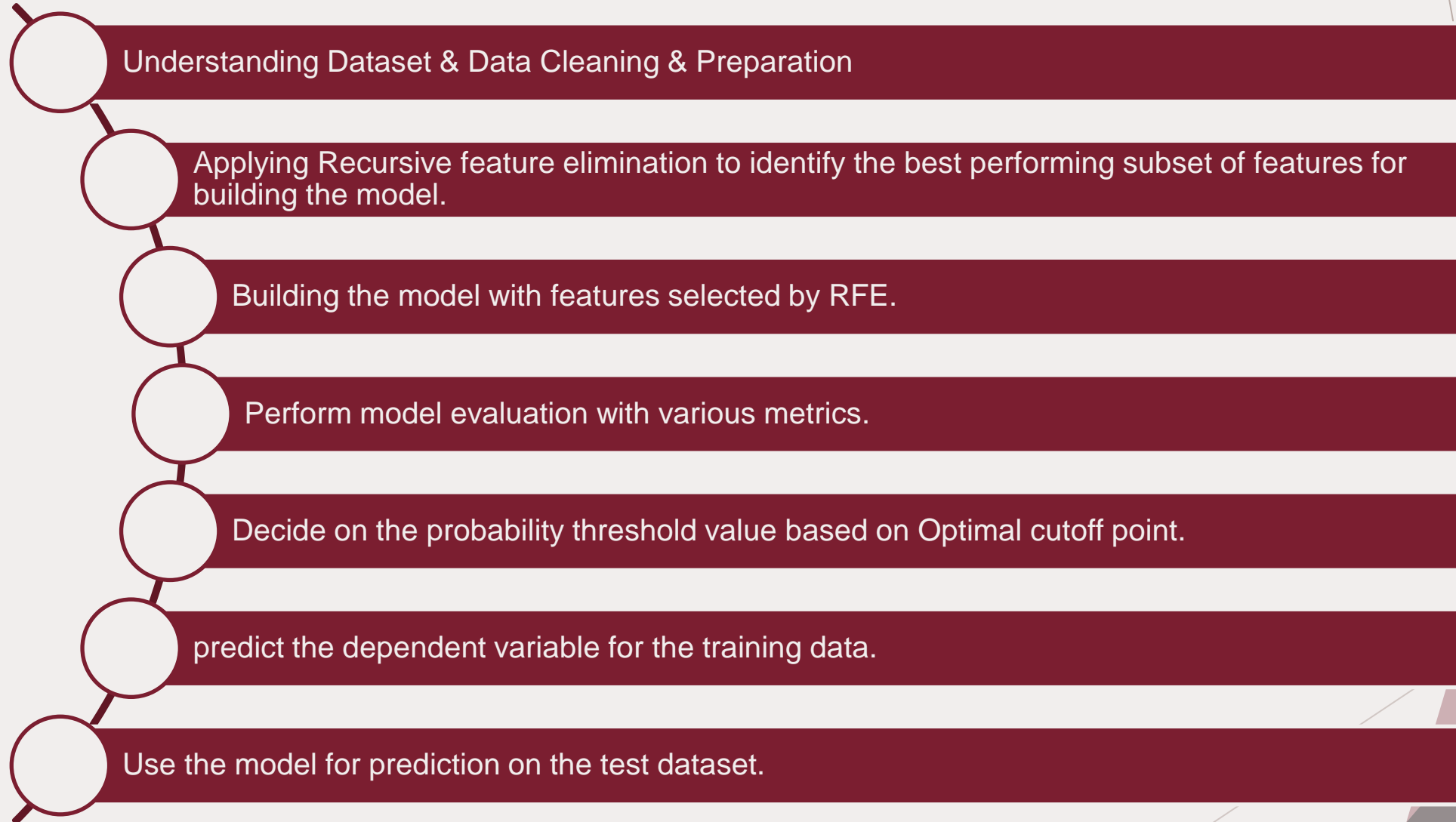
The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with a higher lead score have a higher conversion chance and the customers with a lower lead score have a lower conversion chance. The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%. There are quite a few goals for this case study:

- **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. A higher score would mean that the lead is hot, i.e. is most likely to convert whereas a lower score would mean that the lead is cold and will mostly not get converted.**
- **There are some more problems presented by the company which your model should be able to adjust to if the company's requirement changes in the future so you will need to handle these as well. These problems are provided in a separate doc file. Please fill it based on the logistic regression model you got in the first step. Also, make sure you include this in your final PPT where you'll make recommendations.**

# Problem Solving Methodology



# Problem Solving Methodology



# Data Cleaning & Preparation

- ▶ Data Cleaning
- ▶ Data Preparation



# Data Cleaning

Remove columns which has only one unique value

- Prospect ID, and Lead Number

Handling 'Select' values in some columns

- Specialization 1942
- How did you hear about X Education 5043
- Lead Profile 4146
- City 2249

Categorical Attributes Analysis (Null Values and duplication)

- Country
- Specialization
- What is your current occupation
- What matters most to you in choosing a course
- Tags
- City
- Lead Source
- Last Activity
- Lead Origin
- Do Not Email

# Data Cleaning

## Categorical Attributes Analysis (Drop Attributes)

- Do Not Call
- Search
- Magazine
- Newspaper Article
- X Education Forum
- Digital Advertisement
- Newspaper
- 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

## Numerical Attributes Analysis

- TotalVisits
- Page Views Per Visit
- Total Time Spent on Website

## Assigning a Unique Category to NULL/SELECT values

- Unspecified
- Others

## Binary Encoding

# Data Preparation

## Dummy Encoding

'Lead Quality', 'Asymmetrique Profile Index', 'Asymmetrique Activity Index', 'Tags', 'Lead Profile', 'Lead Origin', 'What is your current occupation', 'Specialization', 'City', 'Last Activity', 'Lead Source', 'Last Notable Activity' and 'Country'

## TestTrain Split

The original dataframe was split into train and test dataset.

## Feature Scaling

Use standard scaler to scale the numerical variables

# Build the Module

- ▶ Feature Selection using RFE
- ▶ Build the Module
- ▶ VIF check
- ▶ Predicted values on the train set
- ▶ Optimal Cutoff Point

# Feature Selection using RFE

**Recursive feature elimination** is an optimization technique for finding the best performing subset of features.

- Lead Source\_Welingak Website
- Last Activity\_Email Bounced
- Last Activity\_SMS Sent
- Tags\_Already a student
- Tags\_Closed by Horizzon
- Tags\_Interested in full time MBA
- Tags\_Interested in other courses
- Tags\_Lost to EINS
- Tags\_Not doing further education
- Tags\_Others', 'Tags\_Ringing
- Tags\_Will revert after reading the email
- Tags\_switched off
- Last Notable Activity\_Modified
- Last Notable Activity\_Olark Chat Conversation

```
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.feature_selection import RFE
```

```
logistic_reg = LogisticRegression()
# running RFE with 15 variables as output
rfe = RFE(logistic_reg, n_features_to_select=15)
rfe = rfe.fit(X_train, y_train)
```

```
rfe.support_
```

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

```
# use the columns select by RFE in the module
col = X_train.columns[rfe.support_]
col
```

```
Index(['Lead Source_Welingak Website', 'Last Activity_Email Bounced',
      'Last Activity_SMS Sent', 'Tags_Already a student',
      'Tags_Closed by Horizzon', 'Tags_Interested in full time MBA',
      'Tags_Interested in other courses', 'Tags_Lost to EINS',
      'Tags_Not doing further education', 'Tags_Others', 'Tags_Ringing',
      'Tags_Will revert after reading the email', 'Tags_switched off',
      'Last Notable Activity_Modified',
      'Last Notable Activity_Olark Chat Conversation'],
      dtype='object')
```

# Building the Model

## Model 1

The P Value for **Last Activity\_Email Bounced** is high, so we can drop it.

## Model 2

The P Value for **Tags\_Interested in full time MBA** is high, so we can drop it.

## Model 3

The P Value for **Tags\_Not doing further education** is high, so we can drop it.

## Model 4

**All P Values are 0.0** and less than 5%.

```
# Building the Model #1
```

```
X_train_sm = sm.add_constant(X_train[col])
logRegm1 = sm.GLM(y_train,X_train_sm, family = sm.families.Binomial())
res = logRegm1.fit()
res.summary()
```

Generalized Linear Model Regression Results

Dep. Variable:	Converted	No. Observations:	6246
Model:	GLM	Df Residuals:	6230
Model Family:	Binomial	Df Model:	15
Link Function:	Logit	Scale:	1.0000
Method:	IRLS	Log-Likelihood:	-1415.2
Date:	Mon, 14 Aug 2023	Deviance:	2830.5
Time:	12:40:52	Pearson chi2:	1.49e+04
No. Iterations:	8	Pseudo R-squ. (CS):	0.5832
Covariance Type:	nonrobust		

	coef	std err	z	P> z	[0.025	0.975]
const	-1.1896	0.073	-16.390	0.000	-1.332	-1.047
Lead Source_Welingak Website	4.0442	0.748	5.408	0.000	2.578	5.510
Last Activity_Email Bounced	-1.2689	0.528	-2.402	0.016	-2.304	-0.233
Last Activity_SMS Sent	2.0815	0.108	19.266	0.000	1.870	2.293
Tags_Already a student	-2.8884	0.585	-4.935	0.000	-4.036	-1.741
Tags_Closed by Horizzon	6.9793	0.721	9.680	0.000	5.566	8.392
Tags_Interested in full time MBA	-1.5293	0.598	-2.559	0.010	-2.701	-0.358
Tags_Interested in other courses	-1.8636	0.376	-4.961	0.000	-2.600	-1.127
Tags_Lost to EINS	5.8212	0.526	11.076	0.000	4.791	6.851
Tags_Not doing further education	-2.7433	1.027	-2.671	0.008	-4.757	-0.730
Tags_Others	-2.2543	0.312	-7.224	0.000	-2.866	-1.643
Tags_Ringing	-3.5160	0.238	-14.787	0.000	-3.982	-3.050
Tags_Will revert after reading the email	4.6530	0.179	26.025	0.000	4.303	5.003
Tags_switched off	-4.3312	0.722	-6.003	0.000	-5.745	-2.917
Last Notable Activity_Modified	-1.8076	0.121	-14.992	0.000	-2.044	-1.571
Last Notable Activity_Olark Chat Conversation	-1.5081	0.399	-3.781	0.000	-2.290	-0.726

# VIF Check

VIF is low and no need for further variables to drop.

## 6.3.5 VIF Check

```
# Check for the VIF values of the feature variables.
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Create a dataframe contain all the feature variables and their respective VIFs
vif = pd.DataFrame()
vif['Features'] = X_train[col].columns
vif['VIF'] = [variance_inflation_factor(X_train[col].values, i) for i in range(X_train[col].shape[1])]
vif['VIF'] = round(vif['VIF'], 2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

	Features	VIF
3	Tags_Closed by Horizon	1.05
5	Tags_Lost to EINS	1.04
0	Lead Source_Welingak Website	1.03
6	Tags_Others	1.03
9	Tags_switched off	1.02
11	Last Notable Activity_Olark Chat Conversation	1.01
4	Tags_Interested in other courses	0.28
2	Tags_Already a student	0.22
8	Tags_Will revert after reading the email	0.10
10	Last Notable Activity_Modified	0.10
7	Tags_Ringing	0.07
1	Last Activity_SMS Sent	0.06

# Predicted values on the train set

## The Predicted values on the Train set.

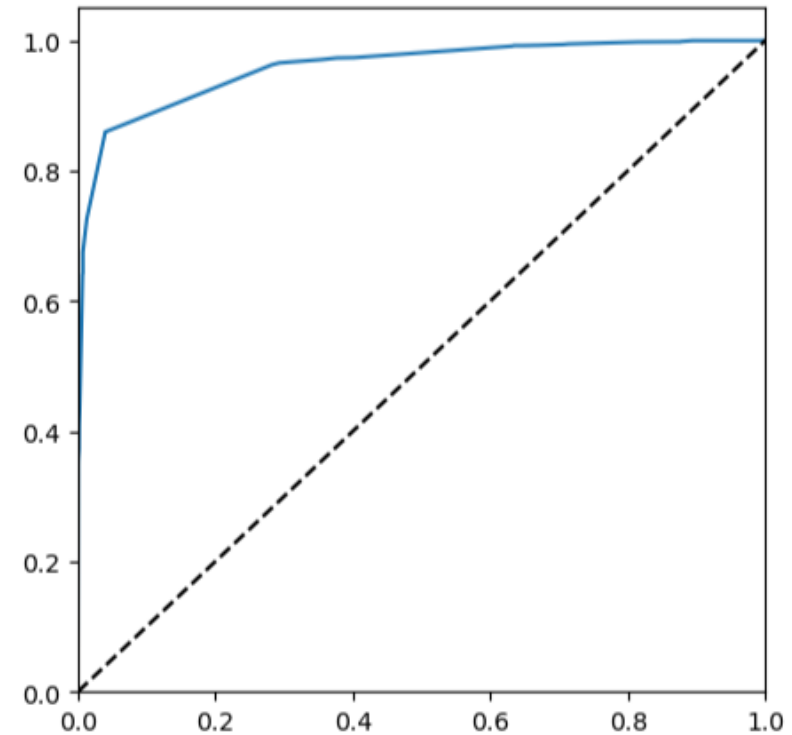
- Overall accuracy 92.11 %
- Sensitivity 85.68 %
- Specificity 96.05 %

## ROC curve

As a rule of thumb, an AUC can be classed as follows,

- 0.90 - 1.00 = excellent
- 0.80 - 0.90 = good
- 0.70 - 0.80 = fair
- 0.60 - 0.70 = poor
- 0.50 - 0.60 = fail

Since we got a value of 0.9582, our model seems to be doing well on the test dataset



### 6.3.6.1 Calculating the Area Under the Curve (GINI)

```
: def auc_val(fpr, tpr):  
    AreaUnderCurve = 0.  
    for i in range(len(fpr)-1):  
        AreaUnderCurve += (fpr[i+1]-fpr[i]) * (tpr[i+1]+tpr[i])  
    AreaUnderCurve *= 0.5  
    return AreaUnderCurve
```

```
: auc = auc_val(fpr, tpr)  
auc
```

```
: 0.9581804239350635
```

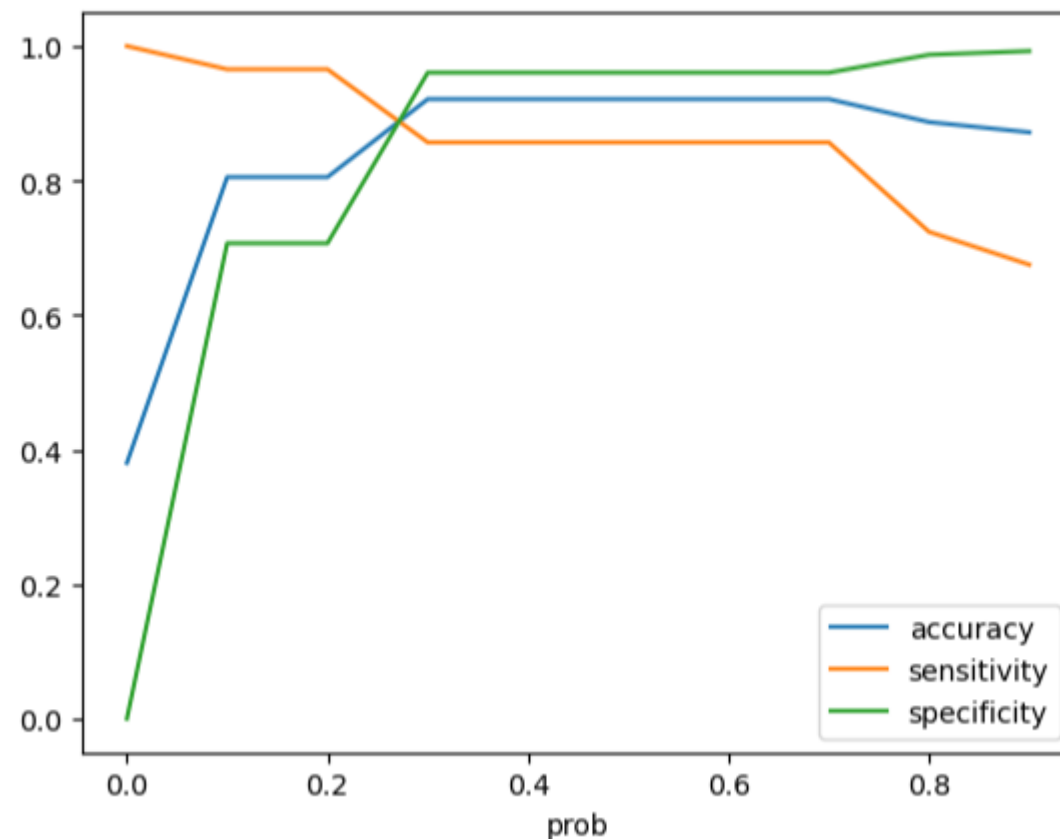


# Optimal Cutoff Point

We can see above the model seems to be performing well. The ROC curve has a value of 0.9582, which is very good. We have the following values for the Train Data:

- Accuracy : 92.11%
- Sensitivity : 85.68%
- Specificity : 96.05%
- F1 Scoure : 89.20%

```
# Let's plot accuracy sensitivity and specificity for various probabilities.  
cutoff_df.plot.line(x='prob', y=['accuracy', 'sensitivity', 'specificity'])  
plt.show()
```



Predictions on test set

The background of the slide is composed of several overlapping, semi-transparent geometric shapes, primarily triangles. These shapes are in various shades of gray and red. A prominent dark red triangle is located in the lower right quadrant. Other lighter gray and reddish triangles are layered behind it and to the left. The overall effect is a modern, abstract design. The text 'Predictions on test set' is positioned on the left side of the slide, centered vertically relative to the main content area.

# Predictions on test set

## ❑ Test Data:

- Accuracy : 93.17%
- Sensitivity : 87.93%
- Specificity : 96.26%
- F1 Scoure : 90.52%

# Feature Importance



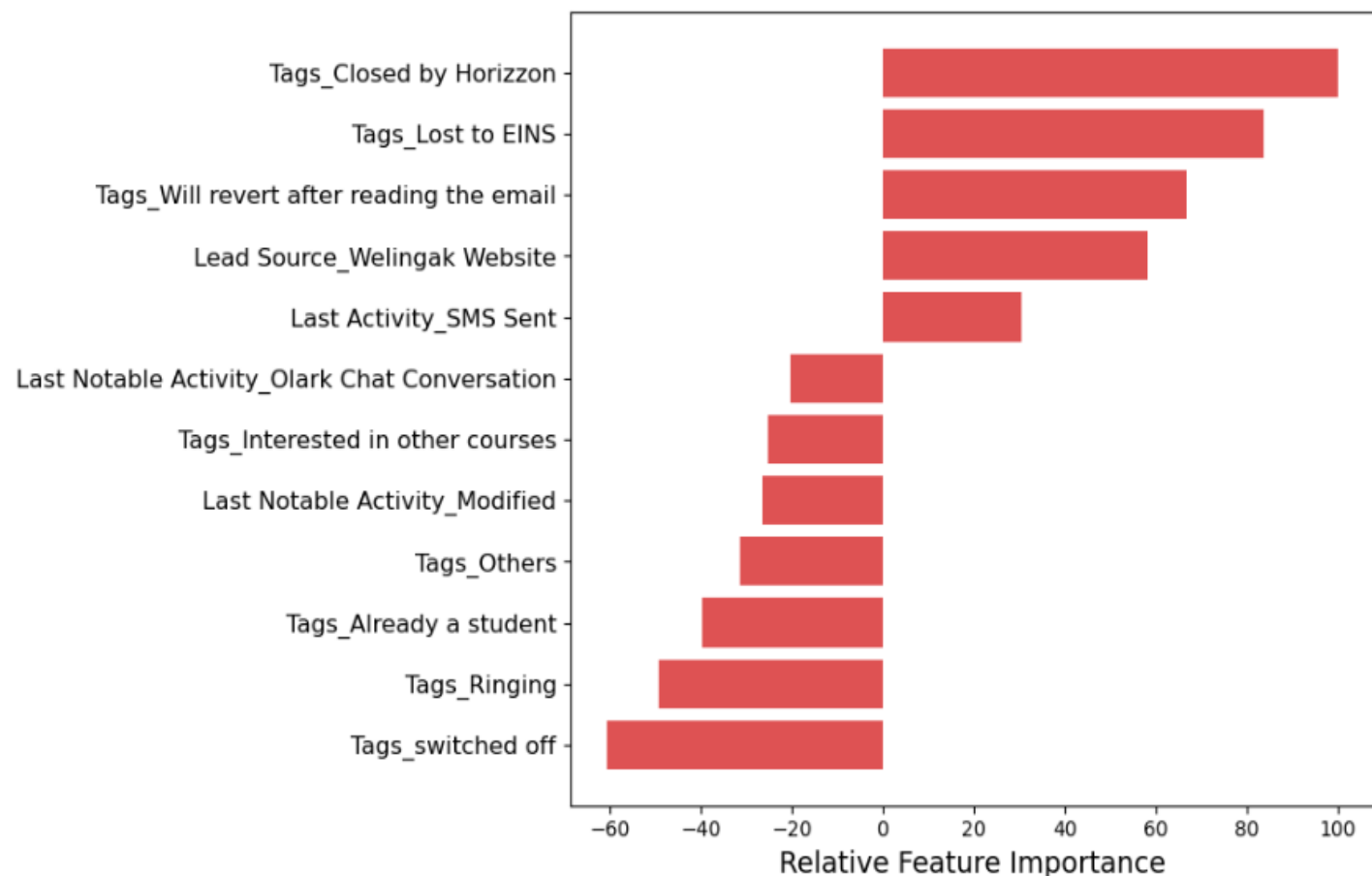
# Feature Importance

	index	0
3	Tags_Closed by Horizon	100.00
5	Tags_Lost to EINS	83.62
8	Tags_Will revert after reading the email	66.83
0	Lead Source_Welingak Website	58.26
1	Last Activity_SMS Sent	30.33
11	Last Notable Activity_Olark Chat Conversation	-20.45
4	Tags_Interested in other courses	-25.19
10	Last Notable Activity_Modified	-26.52
6	Tags_Others	-31.41
2	Tags_Already a student	-39.77
7	Tags_Ringing	-49.15
9	Tags_switched off	-60.69

```
# Plot showing the feature variables based on their relative coefficient values  
pos = np.arange(sorted_idx.shape[0]) + .5
```

```
featfig = plt.figure(figsize=(10,6))  
featax = featfig.add_subplot(1, 1, 1)  
featax.barh(pos, feature_importance[sorted_idx], align='center', color = 'tab:red',alpha=0.8)  
featax.set_yticks(pos)  
featax.set_yticklabels(np.array(X_train[col].columns)[sorted_idx], fontsize=12)  
featax.set_xlabel('Relative Feature Importance', fontsize=14)
```

```
plt.tight_layout()  
plt.show()
```



# Recommendations

The background of the slide is composed of several overlapping, semi-transparent geometric shapes, primarily triangles. These shapes are in various shades of gray, from light to dark, and include some in shades of red and pink. The shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be in front of others. The overall effect is a modern, abstract design.

# Recommendations

## Train Data Result:

- Accuracy : 92.11%
- Sensitivity : 85.68%
- Specificity : 96.05%
- F1 Scoure : 89.20%

## Test Data Result:

- Accuracy : 93.17%
- Sensitivity : 87.93%
- Specificity : 96.26%
- F1 Scoure : 90.52%

**Based on our model, some features are identified which contribute most to a Lead getting converted successfully.**





# Recommendations

The conversion probability of a lead increases with increase in values of the following features in descending order:

- Tags\_Closed by Horizon
- Tags\_Lost to EINS
- Tags\_Will revert after reading the email
- Lead Source\_Welingak Website
- Last Activity\_SMS Sent

The conversion probability of a lead increases with decrease in values of the following features in descending order:

- Last Notable Activity\_Olark Chat Conversation
- Tags\_Interested in other courses
- Last Notable Activity\_Modified
- Tags\_Others
- Tags\_Already a student
- Tags\_Ringing
- Tags\_switched off







Thank you