1. Which are the top three variables in your model which contribute most towards the probability of a lead getting converted?

Ans:

Based on the coefficient values from final model (refer :below screeshot **Fig:a**), the following are the top three variables that contribute most towards the probability of a lead getting converted :

a) Lead Add Form (from Lead Origin)
b) Working Professional (from What is your current Occupation)
c) SMS sent (from Last Notable Activity) & Others (from Last Activity) both coefficeint are almost similar.

Note: We Grouped Last Activity column values like Had a Phone Conversation,Approached upfront,View in browser link Clicked,Email Received,Resubscribed to emails,Visited Booth in Tradeshow into Others as these occurred in small percentage but from co-efficeint has maor effect on lead convertion.

Among these Had a Phone Conversation occurrence percentage is higher compared to other values grouped in Others of Last Activity.

Also difference between coefficeint between Last Notable Activity SMS sent and Last Activity Others is very minimum. Which can also be taken into account for higher lead conversion rate.

Fig: a

	coef	std err	z	P>|z|	[0.025	0.975]
const	-1.7713	0.118	-15.034	0.000	-2.002	-1.540
Do Not Email	-1.4064	0.190	-7.409	0.000	-1.778	-1.034
Total Time Spent on Website	1.0777	0.041	26.215	0.000	0.997	1.158
Lead Origin_Landing Page Submission	-0.2022	0.095	-2.139	0.032	-0.388	-0.017
Lead Origin_Lead Add Form	4.1477	0.232	17.901	0.000	3.694	4.602
Lead Origin_Lead Import	1.2727	0.489	2.602	0.009	0.314	2.231
Lead Source_Google	0.2283	0.085	2.697	0.007	0.062	0.394
Lead Source_Olark Chat	1.1572	0.135	8.599	0.000	0.893	1.421
Last Activity_Email Opened	0.5169	0.094	5.503	0.000	0.333	0.701
Last Activity_Olark Chat Conversation	-1.2298	0.183	-6.711	0.000	-1.589	-0.871
Last Activity_Others	1.9523	0.695	2.808	0.005	0.590	3.315
What is your current occupation_Working Professional	2.5549	0.185	13.825	0.000	2.193	2.917
Last Notable Activity_Others	1.5878	0.297	5.351	0.000	1.006	2.169
Last Notable Activity_SMS Sent	1.9247	0.103	18.680	0.000	1.723	2.127

2. What are the top 3 categorical/dummy variables in the model which should be focused the most on in order to increase the probability of lead conversion?

Ans:

Based on the coefficient values from final model (refer :above screeshot Fig:a), the following are the top three Categorical variables that contribute most towards the probability of a lead getting converted :

a) Lead Add Form (from Lead Origin)
b) Working Professional (from What is your current Occupation)
c) SMS sent (from Last Notable Activity) & Others (from Last Activity) both coefficeint are almost similar.

Note: We Grouped Last Activity column values like Had a Phone Conversation,Approached upfront,View in browser link Clicked,Email Received,Resubscribed to emails,Visited Booth in Tradeshow into Others as these occurred in small percentage but from co-efficeint has maor effect on lead convertion.

Among these Had a Phone Conversation occurrence percentage is higher compared to other values grouped in Others of Last Activity.

Also difference between coefficeint between Last Notable Activity SMS sent and Last Activity Others is very minimum. Which can also be taken into account for higher lead conversion rate.

3. X Education has a period of 2 months every year during which they hire some interns. The sales team, in particular, has around 10 interns allotted to them. So during this phase, they wish to make the lead conversion more aggressive. So they want almost all of the potential leads (i.e. the customers who have been predicted as 1 by the model) to be converted and hence, want to make phone calls to as much of such people as possible. Suggest a good strategy they should employ at this stage.

Ans:

From our model, we know how the Sensitivity and Specificity rating changes with change in the threshold value:

When the probability thresholds are very low, the sensitivity is very high and specificity is very low. Similarly, for larger probability thresholds, the sensitivity values are very low but the specificity values are very high.

High sensitivity implies that our model will correctly identify almost all leads who are even at likelyhood margin to be Convertsion cases. It will do that by over-estimating the Conversion likelihood, i.e. it will misclassify some non-Conversion cases as Conversions.

Now, since X Education has more manpower for these 2 months and they wish to make the lead conversion more aggressive by wanting almost all of the potential leads, we can choose a lower threshold value for Conversion Probability. (More resources available to call more leads)

This will ensure the Sensitivity rating is very high which in turn will make sure almost all leads who are likelyhood margin to Convert are identified correctly and the agents can make phone calls to as much of people as possible for lead conversion.

This can be inferred in the below code output:

When Threshold values is **Optimal**: 0.35

Number of possible leads are : 3452

In [455]:
```
# Let us make the final prediction using 0.35 as the cut off based on above observation

y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.35 else 0)
y_train_pred_final.head()

#displaying the 10 rows with the calculated Lead_score column highlighted
def highlight_cols(s):
    color = 'yellow'
    return 'background-color: %s' % color

y_train_pred_final.head(10).style.applymap(highlight_cols, subset=pd.IndexSlice[:, ['final_predicted']])
```

Out[455]:

	Converted	Converted_Prob	LeadId	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.019386	5279	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0.362031	3099	0	1	1	1	1	0	0	0	0	0	0	1
2	1	0.782688	91	1	1	1	1	1	1	1	1	1	0	0	1
3	1	0.876579	1577	1	1	1	1	1	1	1	1	1	1	0	1
4	0	0.155583	487	0	1	1	0	0	0	0	0	0	0	0	0
5	0	0.147375	758	0	1	1	0	0	0	0	0	0	0	0	0
6	0	0.143573	7260	0	1	1	0	0	0	0	0	0	0	0	0
7	0	0.657744	7538	1	1	1	1	1	1	1	1	0	0	0	1
8	0	0.109164	2404	0	1	1	0	0	0	0	0	0	0	0	0
9	0	0.966695	8190	1	1	1	1	1	1	1	1	1	1	1	1

In [456]:
```
#for subjective question
print('Total Number of possible leads 1/  Total Number of leads not possible 0')
y_train_pred_final.final_predicted.value_counts()
```

```
Total Number of possible leads 1/  Total Number of leads not possible 0
```

Out[456]:
```
0    3452
1    2459
Name: final_predicted, dtype: int64
```

When Threshold values is **reduced** : 0.1 :

Number of possible leads are : 4472

```
[457]: #for subjective question
y_train_pred_final['final_predicted2'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.1 else 0)
y_train_pred_final.head()

#displaying the 10 rows with the calculated Lead_score column highlighted
def highlight_cols(s):
    color = 'yellow'
    return 'background-color: %s' % color

y_train_pred_final.head(10).style.applymap(highlight_cols, subset=pd.IndexSlice[:, ['final_predicted2']])
```

[457]:

	Converted	Converted_Prob	LeadId	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted	final_predicted2
0	0	0.019386	5279	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0.362031	3099	0	1	1	1	1	0	0	0	0	0	0	1	1
2	1	0.782688	91	1	1	1	1	1	1	1	1	1	0	0	1	1
3	1	0.876579	1577	1	1	1	1	1	1	1	1	1	1	0	1	1
4	0	0.155583	487	0	1	1	0	0	0	0	0	0	0	0	0	1
5	0	0.147375	758	0	1	1	0	0	0	0	0	0	0	0	0	1
6	0	0.143573	7260	0	1	1	0	0	0	0	0	0	0	0	0	1
7	0	0.657744	7538	1	1	1	1	1	1	1	1	0	0	0	1	1
8	0	0.109164	2404	0	1	1	0	0	0	0	0	0	0	0	0	1
9	0	0.986695	8190	1	1	1	1	1	1	1	1	1	1	1	1	1

```
[458]: #for subjective question
print('Total Number of possible leads 1/  Total Number of leads not possible 0')
y_train_pred_final.final_predicted2.value_counts()
```

```
Total Number of possible leads 1/  Total Number of leads not possible 0
```

```
[458]: 1    4472
0    1439
Name: final_predicted2, dtype: int64
```

4. Similarly, at times, the company reaches its target for a quarter before the deadline. During this time, the company wants the sales team to focus on some new work as well. So during this time, the company's aim is to not make phone calls unless it's extremely necessary, i.e. they want to minimize the rate of useless phone calls. Suggest a strategy they should employ at this stage.

Ans:

From our model, we know how the Sensitivity and Specificity rating changes with change in the cut-off threshold value.

When the probability thresholds are very high, the sensitivity is very low and specificity is very high. Similarly, for lower probability thresholds, the sensitivity values are very high but the specificity values are very low.

High specificity implies that our model will not correctly identify almost all leads who are at the likelyhood margin to Convertsion cases, it will be misclassified as non-Conversions. Only the absolute leads which can be converted are classified as lead conversions.

Now, since X Education has more important work other than lead conversion , they wish to make the lead conversion minimal by investing only less time on lead conversion. we can choose a Higher threshold value for Conversion Probability.

This will ensure the specificity rating is very high which in turn will make sure who are absolutly to Convert are only identified and the agents can make phone calls to such people who as Absolute lead leading to minimum number of calls.

This can be inferred in the below code output:

When Threshold values is **Optimal**: 0.35

Number of possible leads are : 3452

In [455]:
```
# Let us make the final prediction using 0.35 as the cut off based on above observation

y_train_pred_final['final_predicted'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.35 else 0)
y_train_pred_final.head()

#displaying the 10 rows with the calculated Lead_score column highlighted
def highlight_cols(s):
    color = 'yellow'
    return 'background-color: %s' % color

y_train_pred_final.head(10).style.applymap(highlight_cols, subset=pd.IndexSlice[:, ['final_predicted']])
```

Out[455]:

	Converted	Converted_Prob	LeadId	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted
0	0	0.019386	5279	0	1	0	0	0	0	0	0	0	0	0	0
1	0	0.362031	3099	0	1	1	1	1	0	0	0	0	0	0	1
2	1	0.782688	91	1	1	1	1	1	1	1	1	1	0	0	1
3	1	0.876579	1577	1	1	1	1	1	1	1	1	1	1	0	1
4	0	0.155583	487	0	1	1	0	0	0	0	0	0	0	0	0
5	0	0.147375	758	0	1	1	0	0	0	0	0	0	0	0	0
6	0	0.143573	7260	0	1	1	0	0	0	0	0	0	0	0	0
7	0	0.657744	7538	1	1	1	1	1	1	1	1	1	0	0	1
8	0	0.109164	2404	0	1	1	0	0	0	0	0	0	0	0	0
9	0	0.966695	8190	1	1	1	1	1	1	1	1	1	1	1	1

In [456]:
```
#for subjective question
print('Total Number of possible leads 1/  Total Number of leads not possible 0')
y_train_pred_final.final_predicted.value_counts()
```

```
Total Number of possible leads 1/  Total Number of leads not possible 0
```

Out[456]:
```
0    3452
1    2459
Name: final_predicted, dtype: int64
```

When Threshold values is **Increased** : 0.7

Number of possible leads are : 1279

In [753]:
```
#for subjective question
y_train_pred_final['final_predicted2'] = y_train_pred_final.Converted_Prob.map( lambda x: 1 if x > 0.7 else 0)
y_train_pred_final.head()

#displaying the 10 rows with the calculated Lead_score column highlighted
def highlight_cols(s):
    color = 'yellow'
    return 'background-color: %s' % color

y_train_pred_final.head(10).style.applymap(highlight_cols, subset=pd.IndexSlice[:, ['final_predicted2']])
```

Out[753]:

	Converted	Converted_Prob	LeadId	predicted	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	final_predicted	final_predicted2
0	0	0.019386	5279	0	1	0	0	0	0	0	0	0	0	0	0	0
1	0	0.362031	3099	0	1	1	1	1	0	0	0	0	0	0	1	0
2	1	0.782688	91	1	1	1	1	1	1	1	1	1	0	0	1	1
3	1	0.876579	1577	1	1	1	1	1	1	1	1	1	1	0	1	1
4	0	0.155583	487	0	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0.147375	758	0	1	1	0	0	0	0	0	0	0	0	0	0
6	0	0.143573	7260	0	1	1	0	0	0	0	0	0	0	0	0	0
7	0	0.657744	7538	1	1	1	1	1	1	1	1	0	0	0	1	0
8	0	0.109164	2404	0	1	1	0	0	0	0	0	0	0	0	0	0
9	0	0.966695	8190	1	1	1	1	1	1	1	1	1	1	1	1	1

In [754]:
```
#for subjective question
print('Total Number of possible leads 1/  Total Number of leads not possible 0')
y_train_pred_final.final_predicted2.value_counts()
```

```
Total Number of possible leads 1/  Total Number of leads not possible 0
```

Out[754]:
```
0    4632
1    1279
Name: final_predicted2, dtype: int64
```