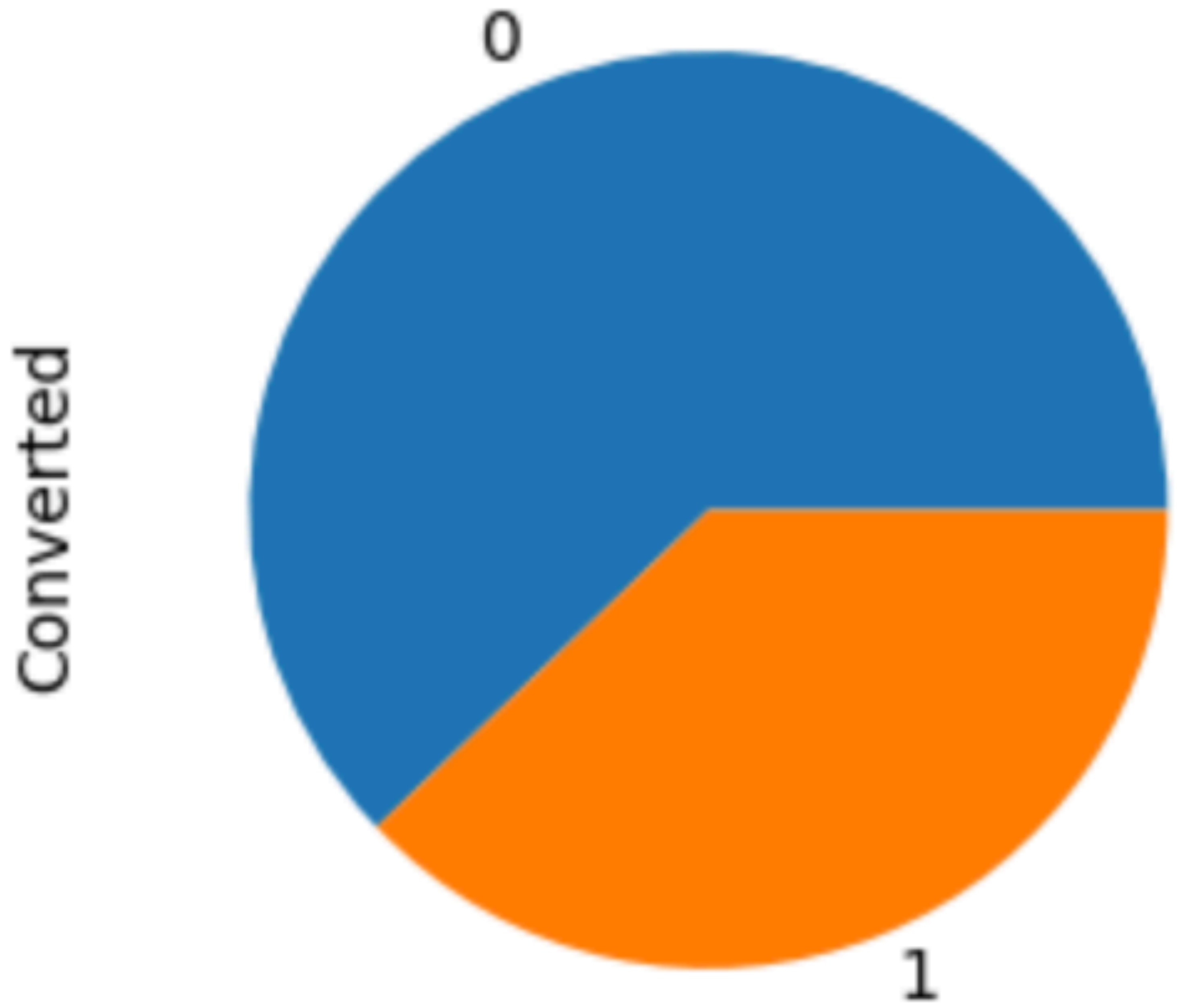
Lead Scoring Case Study

Challenge

The company sells online courses to a varied audience

The conversion ratio is around 30%



Goal

Increase the Conversion ratio

Use the company's data to get insights out of it

Build a logistic regression model to provide lead scores to prospective leads

Approach

Tackling Missing values

Dealing with outliers

Dropping useless columns

EDA (Univariate and Bivariate Analysis)

Collating categories together

Creating Dummy Variables

Splitting data into train and test sets and Feature Scaling

Dropping features using Recursive Feature Elimination, p-value and variance inflation factor

Building the final Logistic Regression model

Tackling Missing Values

A lot of columns had the ‘Select’ keyword which were essentially missing values

```
df.replace('select', np.nan, inplace = True)
```

When number of missing values were very low, we dropped them

```
print(df['Lead Source'].value_counts(normalize = True, dropna = False) * 100)
# Only 0.38% of the values in this column are missing
# We can drop these values as they won't have any effect on our analysis as we have approximately 99.6% of the va
```

When number of missing values were too high, we dropped the columns

```
df['How did you hear about X Education'].value_counts(normalize = True, dropna = False) * 100
# 78% of the values in this column are missing
# This column has no useful information for us to get insights from
# We should drop this column
```

In some special cases, we imputed some value in missing values’ place

```
df['Specialization'].value_counts(normalize = True, dropna = False) * 100
# 36% of the values in this column are not provided
# But these values aren't actually missing, they do have information hidden in them
# There might be the case that the students didn't find the Specialization they were looking for,
# or they were just not sure at the moment to make a decision and decided to not make a decision
# So it would be wise to take these missing values as "unstated", that would be a more correct representation
```

```
df['What is your current occupation'].value_counts(normalize = True, dropna = False) * 100
# We should impute 'Unemployed' in the case of missing values in this case
```

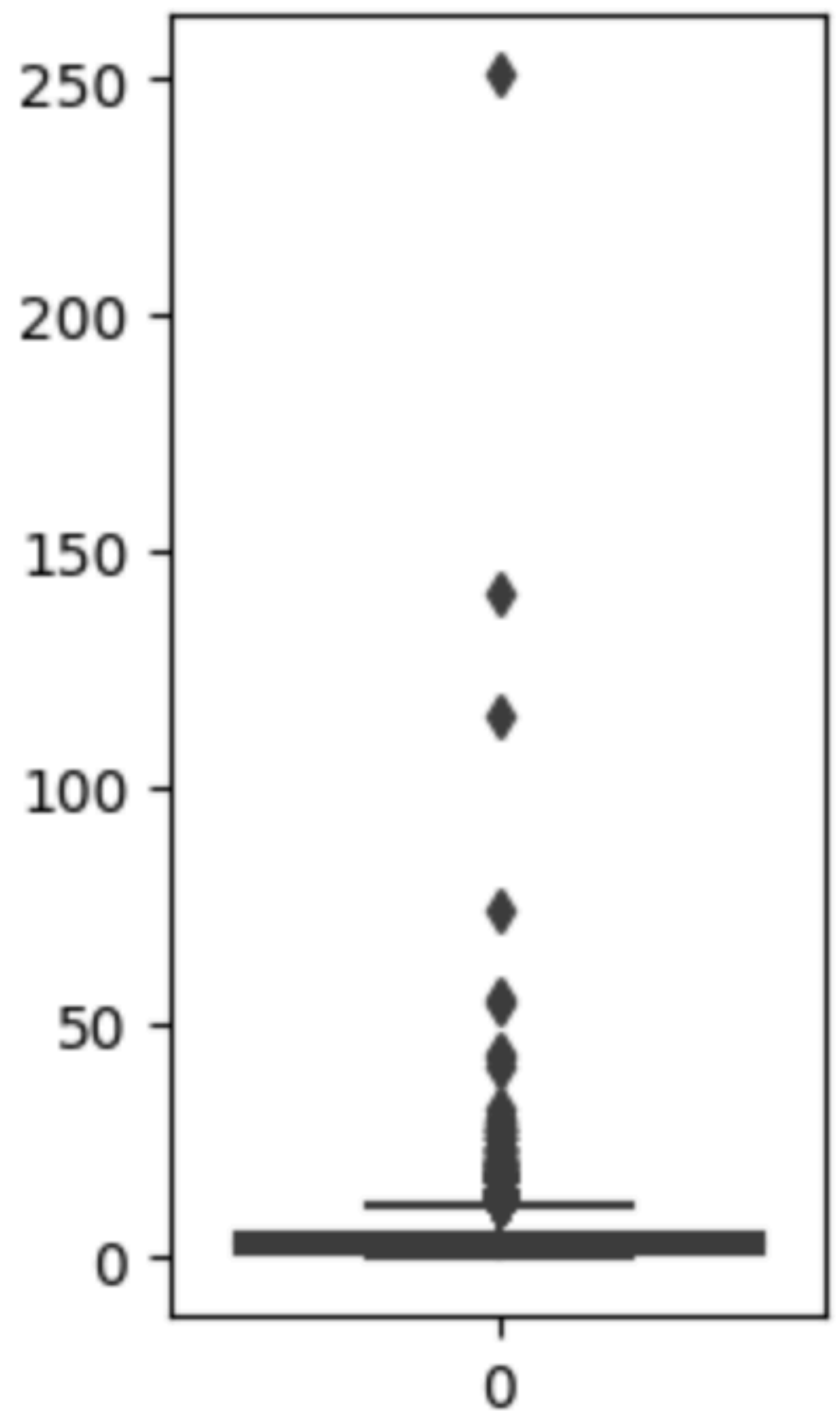
```
unemployed              60.348248
NaN                     29.567996
working professional     7.460877
student                  2.270223
other                    0.165307
housewife                0.099184
businessman              0.088164
Name: What is your current occupation, dtype: float64
```

```
df['What is your current occupation'].replace(np.nan, "unemployed", inplace = True)
```

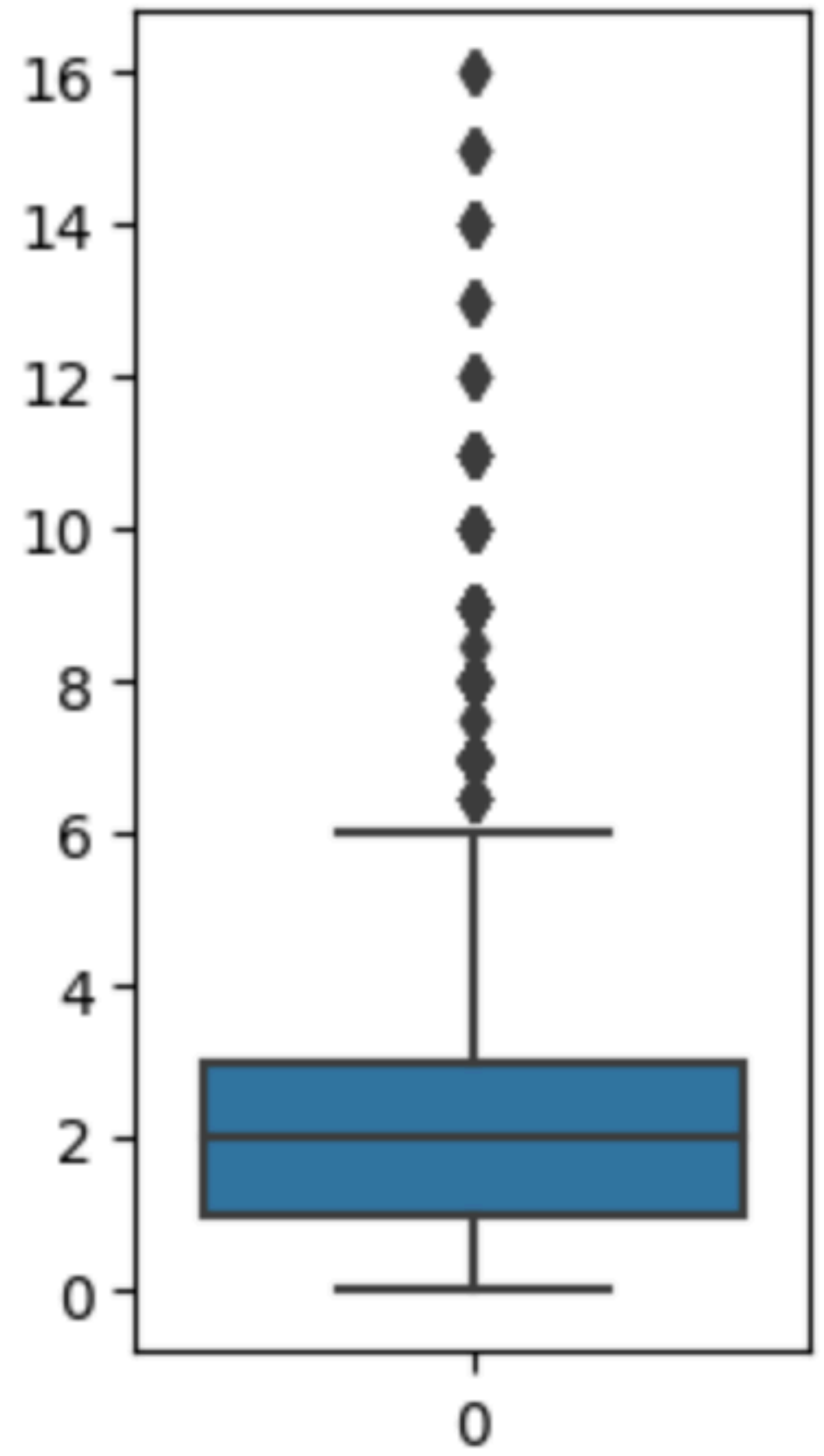
Dealing with Outliers

In both of these cases, we retained the values till the 99th percentile

```
plt.figure(figsize = (2,4))
sns.boxplot(df['TotalVisits'])
plt.show()
# This column has a lot of outliers
# We need to remove them accordingly
```




```
plt.figure(figsize = (2,4))
sns.boxplot(df['Page Views Per Visit'])
plt.show()
# This column has a couple of outliers
```



Dropping Useless Columns

There were some columns which didn't have the necessary amount of variance to be conducive to our analysis and making predictions.

We dropped them.

Column: 'Do Not Email'

```
df['Do Not Email'].value_counts(normalize = True) * 100
# Over 90% of values are from one category
# This column does not have the necessary amount of variance to be conducive to our analysis and making predicti
# It would be wise to drop this column
```

```
no     92.109323
yes     7.890677
Name: Do Not Email, dtype: float64
```

```
df.drop('Do Not Email', axis = 1, inplace = True)
```

Column: 'Do Not Call'

```
df['Do Not Call'].value_counts(normalize = True) * 100
# Over 99% of values are from one category
# This column does not have the necessary amount of variance to be conducive to our analysis and making predicti
# It would be wise to drop this column
```

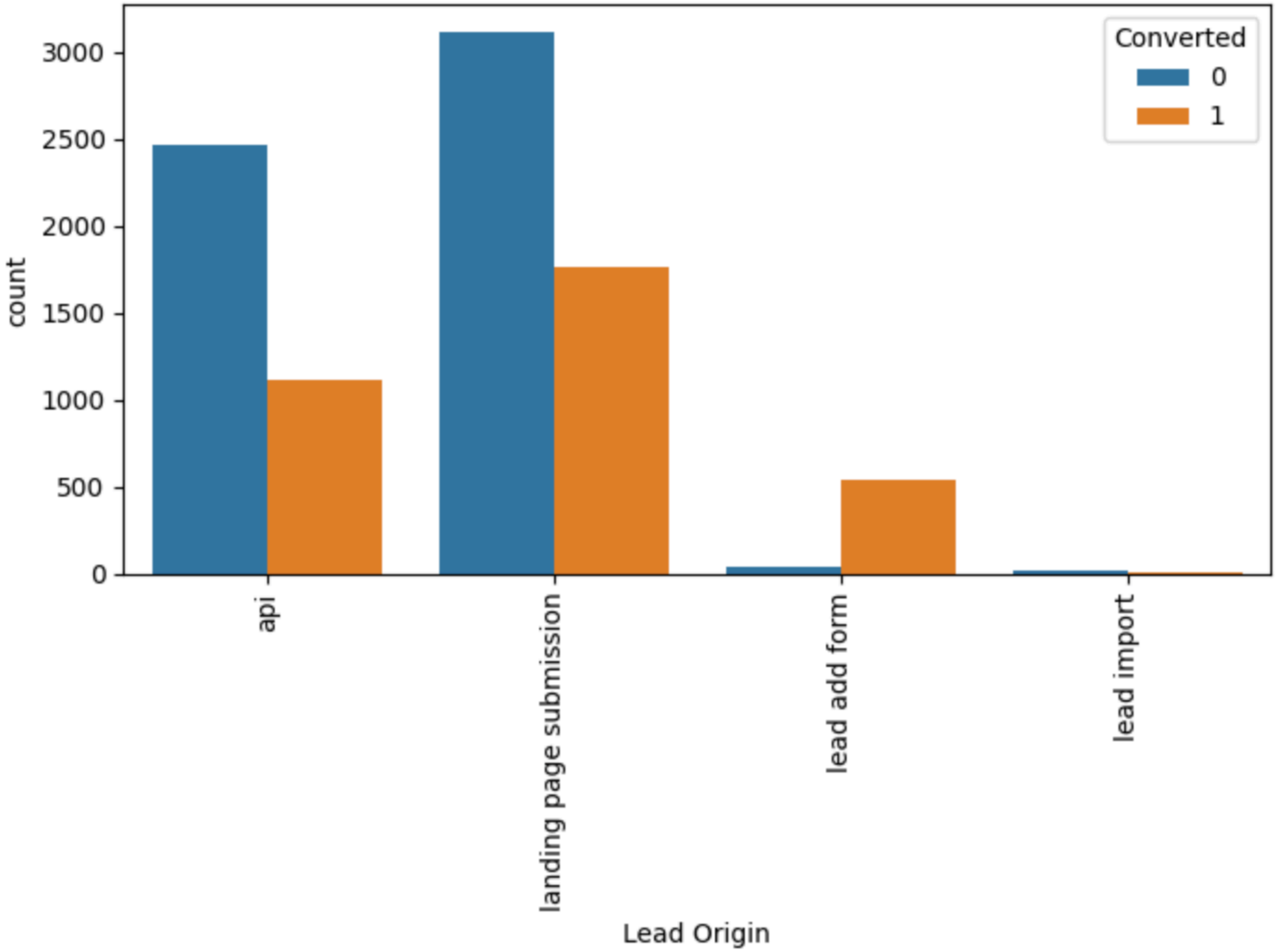
```
no     99.977959
yes     0.022041
Name: Do Not Call, dtype: float64
```

```
df.drop('Do Not Call', axis = 1, inplace = True)
```

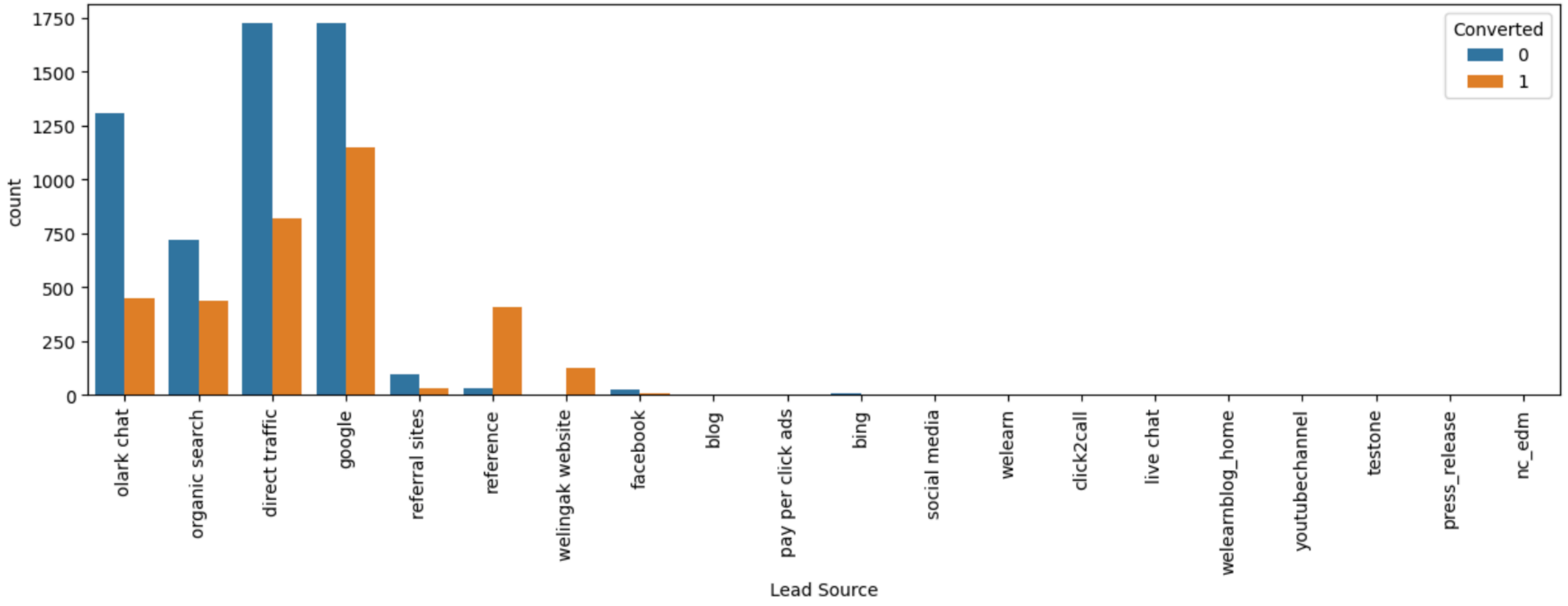
```
-------------------------------------------
no     99.854325
yes     0.145675
Name: Search, dtype: float64
-------------------------------------------
no    100.0
Name: Magazine, dtype: float64
-------------------------------------------
no     99.988794
yes     0.011206
Name: Newspaper Article, dtype: float64
-------------------------------------------
no    100.0
Name: X Education Forums, dtype: float64
-------------------------------------------
no     99.988794
yes     0.011206
Name: Newspaper, dtype: float64
-------------------------------------------
no     99.966383
yes     0.033617
Name: Digital Advertisement, dtype: float64
-------------------------------------------
no     99.932766
yes     0.067234
Name: Through Recommendations, dtype: float64
-------------------------------------------
no    100.0
Name: Receive More Updates About Our Courses, dtype: float64
-------------------------------------------
no    100.0
Name: Update me on Supply Chain Content, dtype: float64
-------------------------------------------
no    100.0
Name: Get updates on DM Content, dtype: float64
-------------------------------------------
no    100.0
Name: I agree to pay the amount through cheque, dtype: float64
-------------------------------------------
```

Exploratory Data Analysis

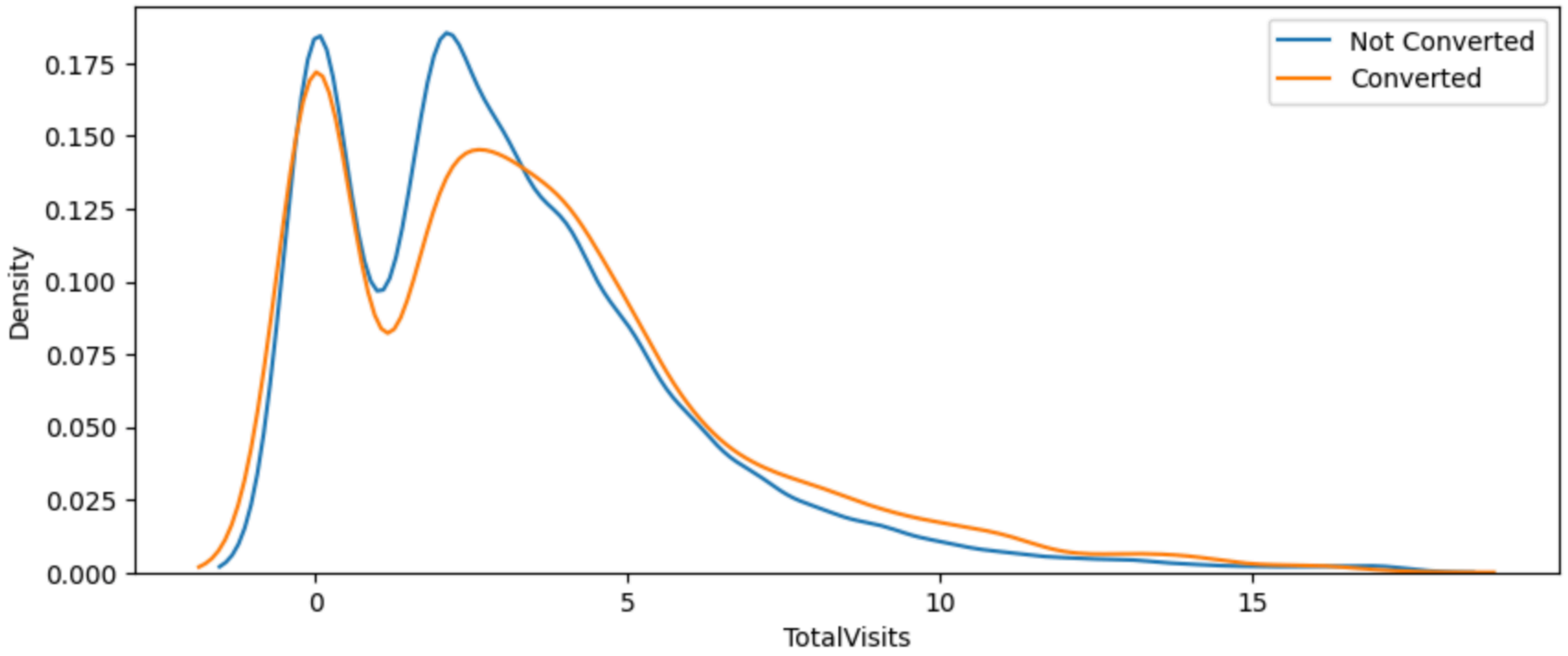
```
# Insights:
# Majority of the Leads come from 'api' and 'landing page submission'
# Majority of the conversions also come from these two but the ratio is not very impressive
# A minority of the Lead come from 'lead add form' but the conversion rate is very impressive
# 'lead import' doesn't perform really well

# Conclusion:
# Try improving the conversion ratio in the 'api' and 'landing page submission' categories
# Take advantage of the leads coming from 'lead add form' and try to increase the numbers
```



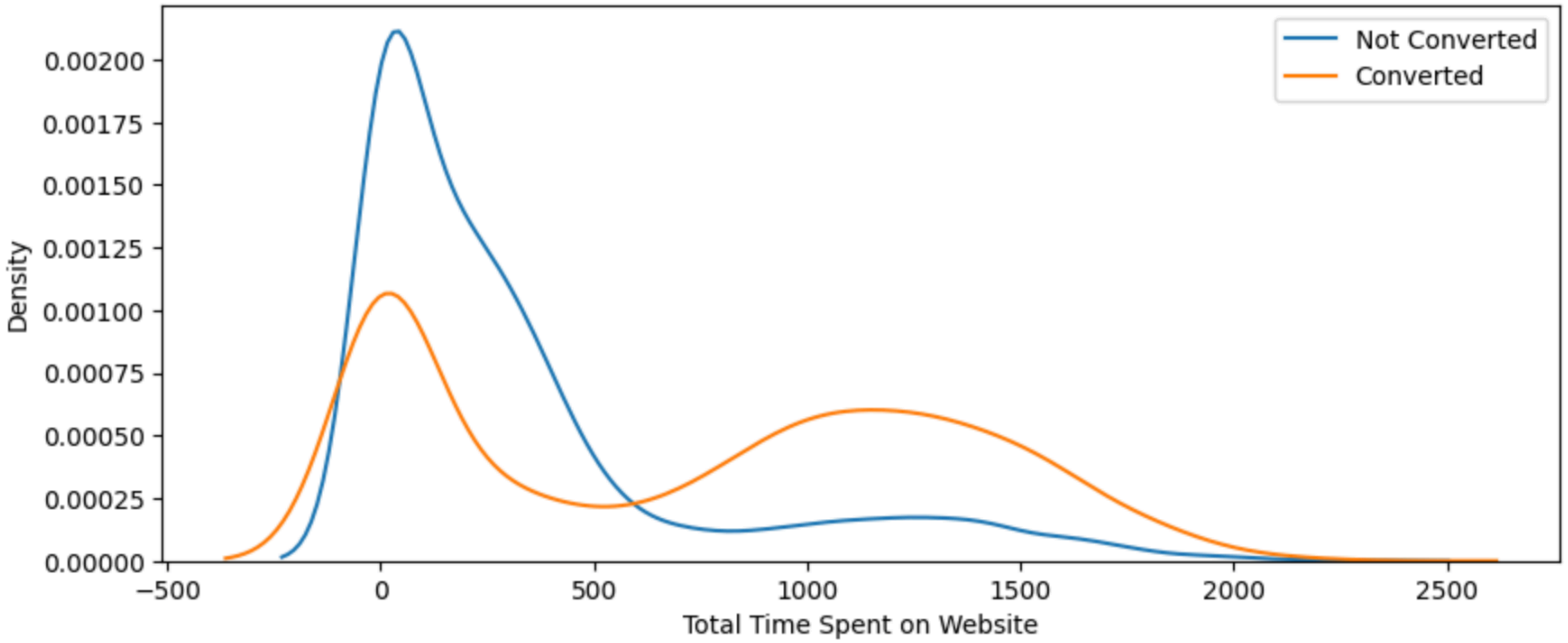

```
plt.figure(figsize=(15,4))
s1 = sns.countplot(data = df, x = 'Lead Source', hue = 'Converted')
s1.set_xticklabels(s1.get_xticklabels(), rotation=90)
plt.show()

# Majority of the leads come from 'olark chat', 'organic search', 'direct traffic', 'google'
# Majority of the converted leads also come from these categories due to their numbers but again, the conversion
# 'referral sites' and 'facebook' perform really badly when it comes to converted leads
# On the other hand, 'reference' and 'welingak website' perform really good when it comes to conversion

# Conclusion:
# Try to improve the conversion rate if possible in the top four ('olark chat' etc) categories mentioned above
# Take advantage (i.e. get more number of leads) of the 'reference' and 'welingak website' categories, they perfo
```



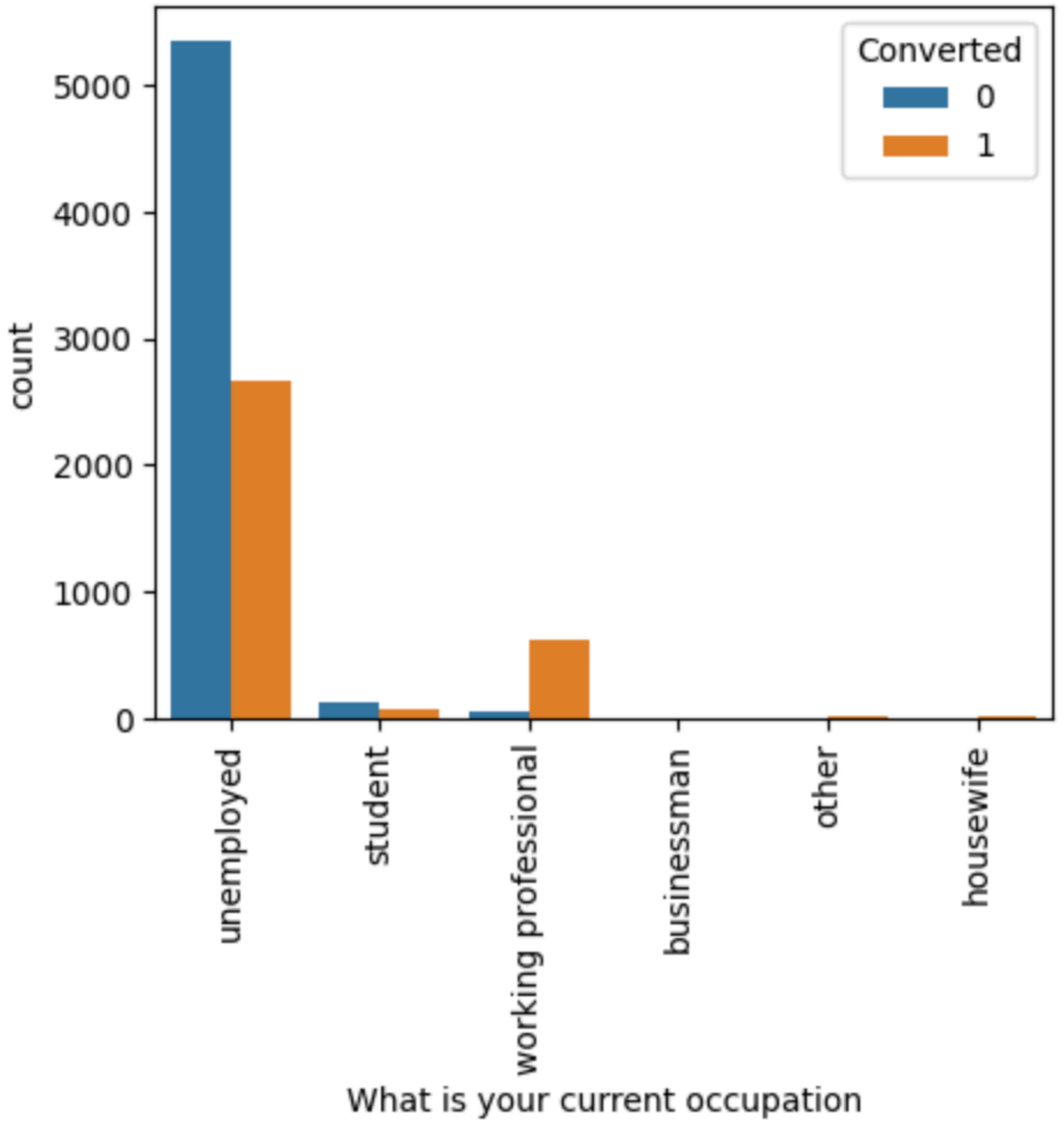
```
# We can definitely see a trend here
# Leads convert when TotalVisits are either low (around 0-3) and high (around 7-15)
# It basically says that there are two types of people who buy the product:
# One who visits the site a couple of times and others who visit a lot. In the middle, we have people where it To
```

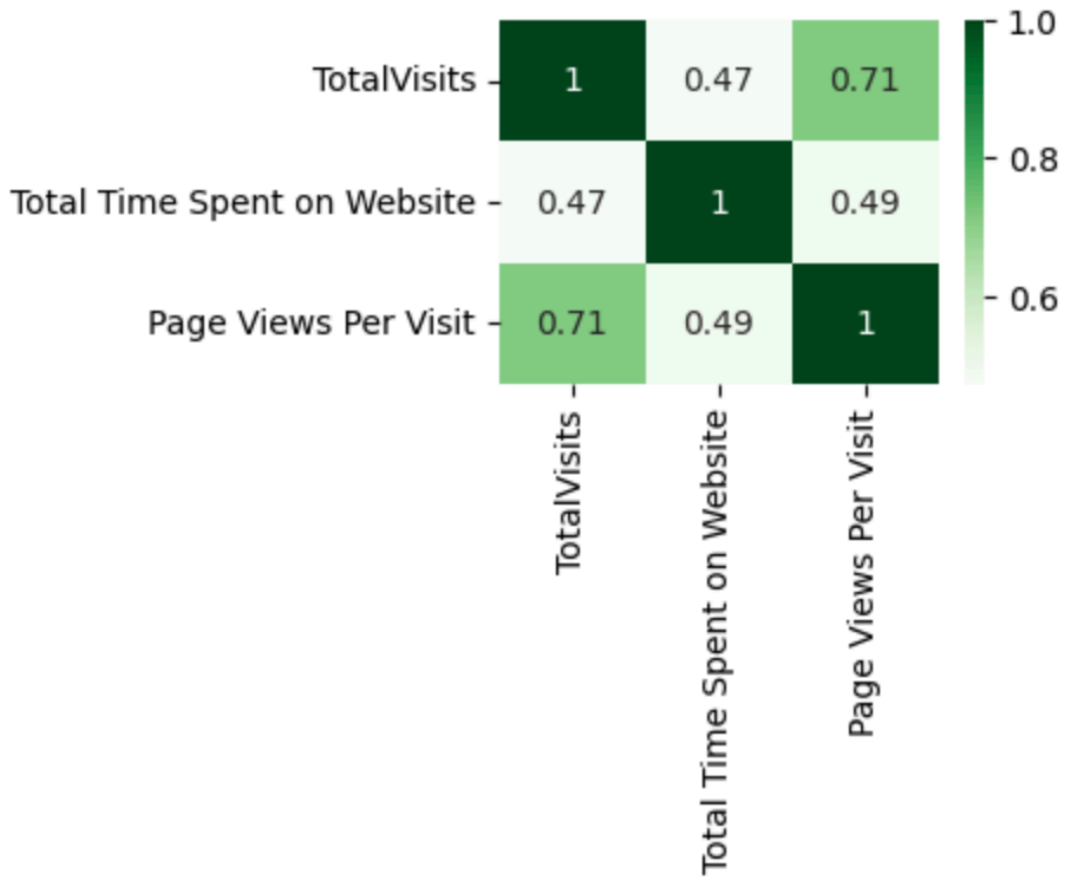


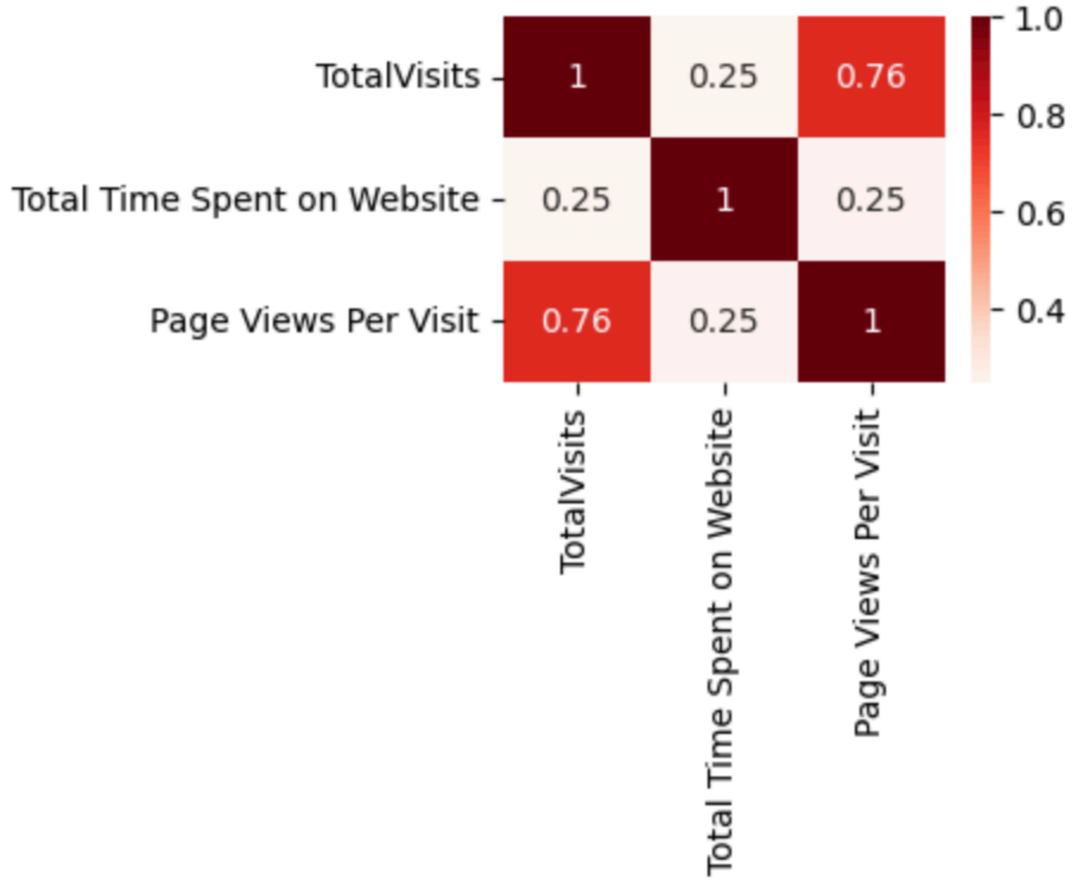

```
# There is a very clear trend here
# For values above 500 (approximately), we can see that leads do convert
# It makes sense since people who spent more time on the website were probably serious about the product
```




```
# Insight:
# A sharp trend can be seen
# 'unemployed' gets the most number of leads but the conversion ratio is bad
# 'working professional' gets less number of lead but the conversion ratio is pretty good

# Conclusion:
# Try to convert more leads in the 'unemployed' category
# Take advantage of the 'working professional' category, the conversion ratio is remarkable
```

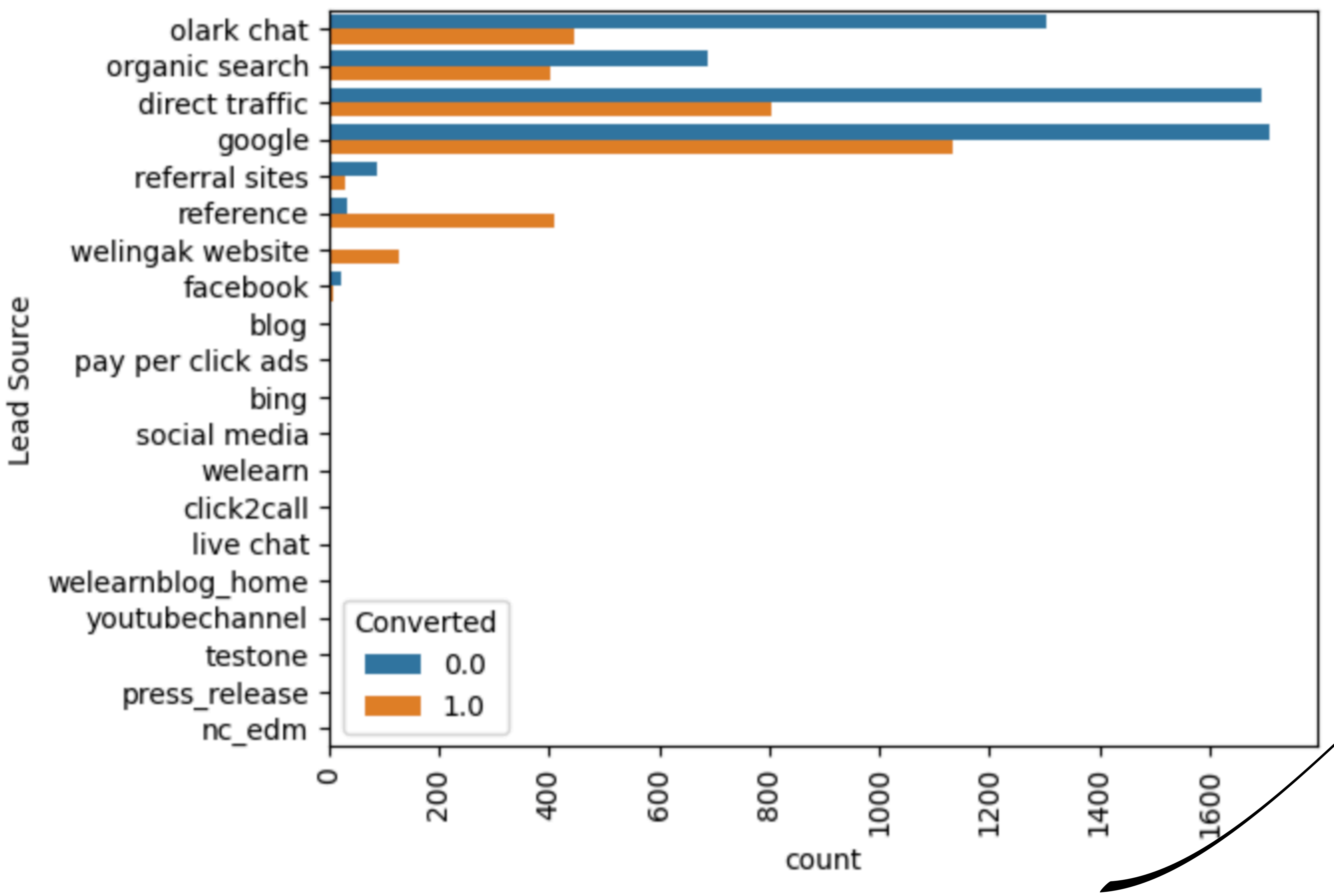


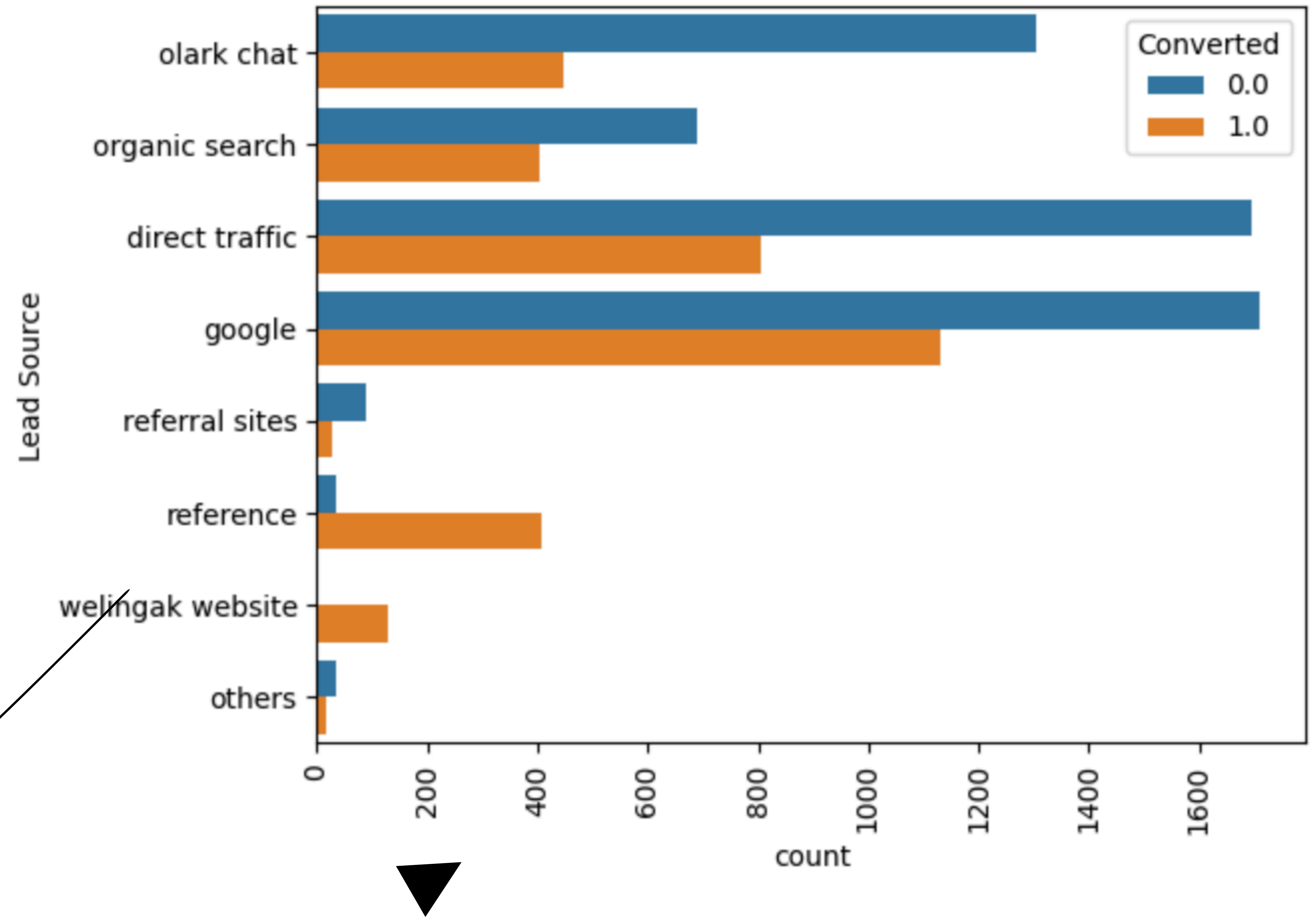


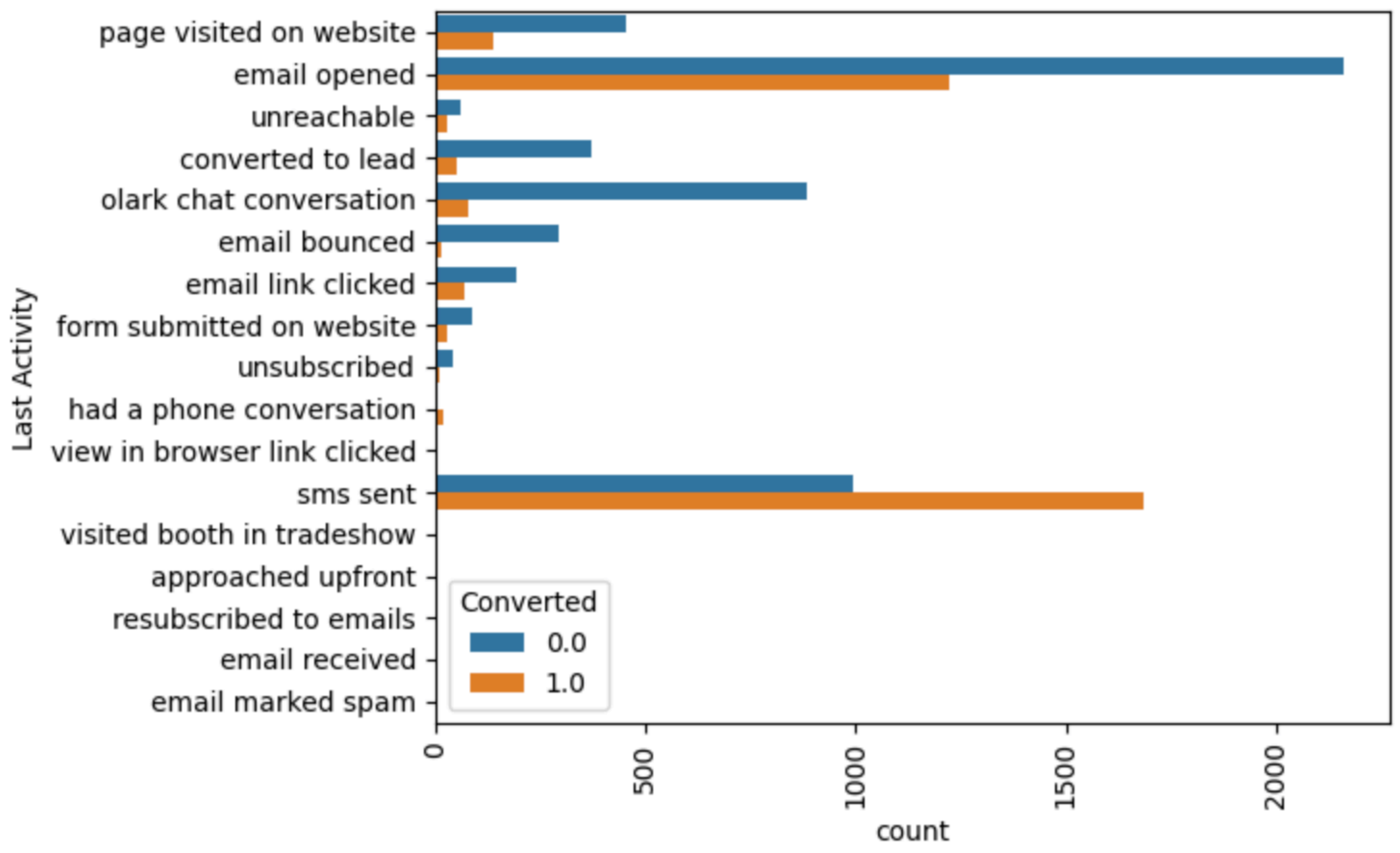


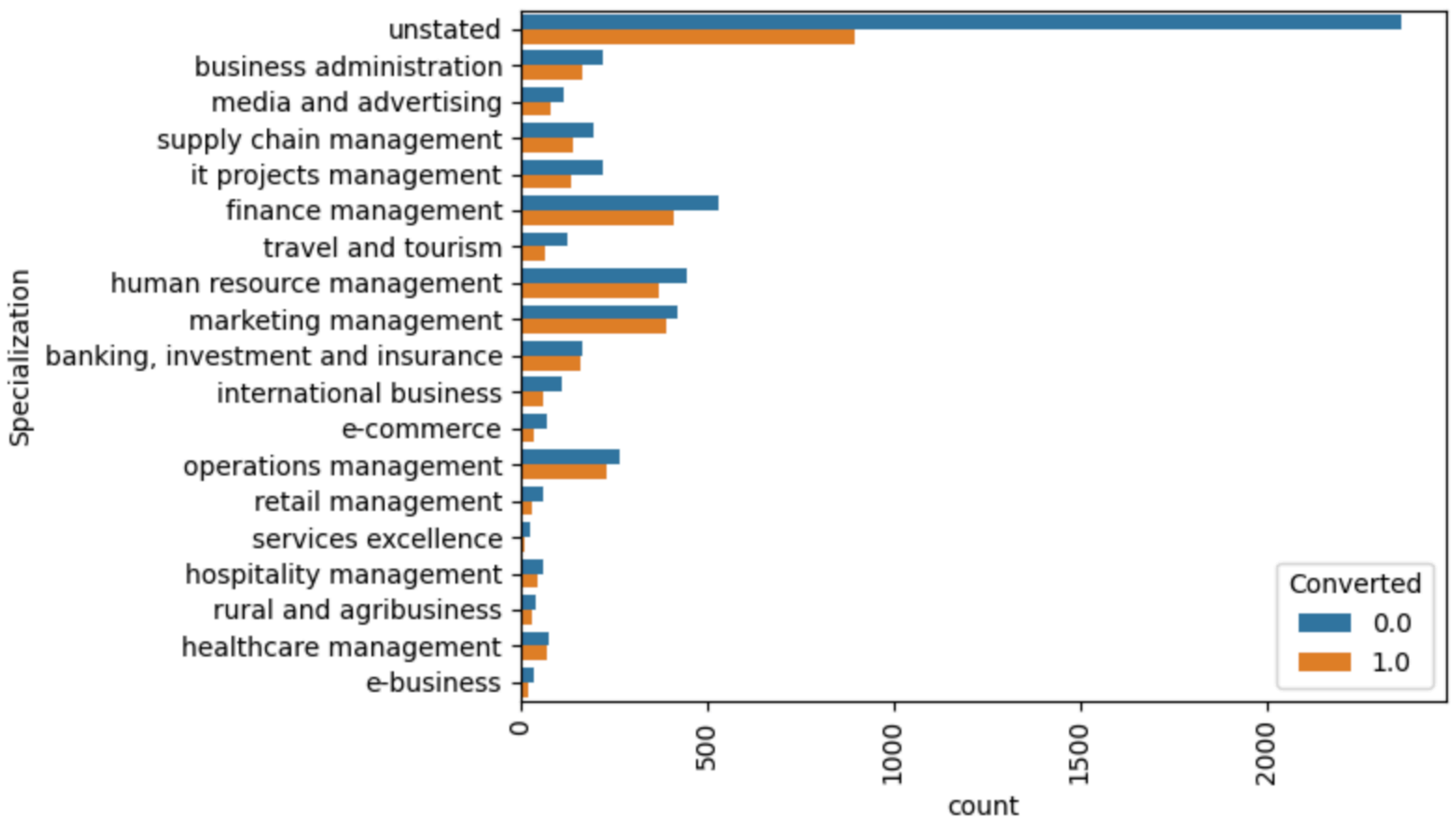
Collating categories together

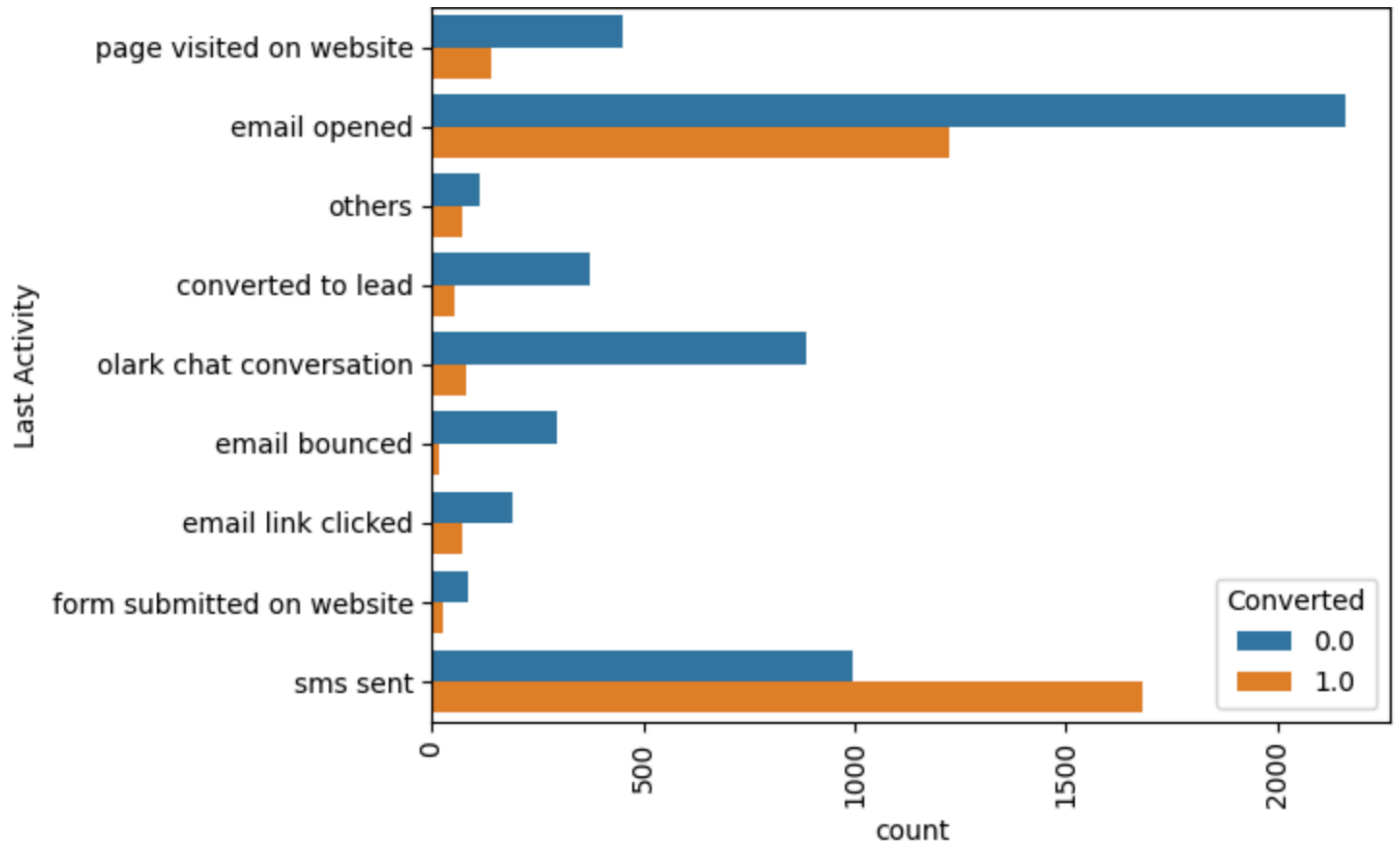
Collating categories which had low conversion impact was helpful in reducing dimensionality when creating dummy variables

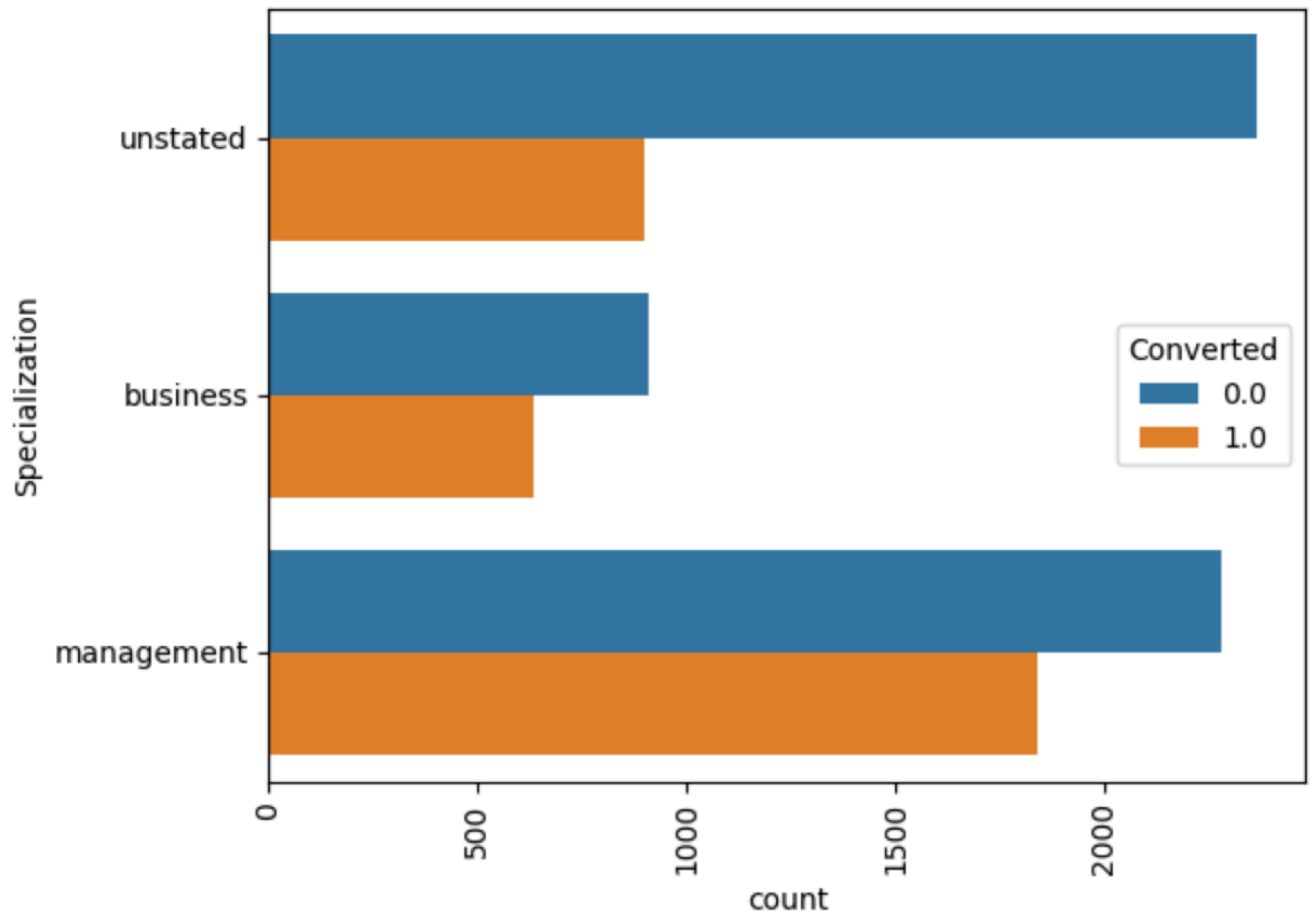





Last Activity
page visited on website
email opened
unreachable
converted to lead
olark chat conversation
email bounced
email link clicked
form submitted on website
unsubscribed
had a phone conversation
view in browser link clicked
sms sent
visited booth in tradeshow
approached upfront
resubscribed to emails
email received
email marked spam
Converted
0.0
1.0
0
500
1000
1500
2000
count


Specialization
unstated
business administration
media and advertising
supply chain management
it projects management
finance management
travel and tourism
human resource management
marketing management
banking, investment and insurance
international business
e-commerce
operations management
retail management
services excellence
hospitality management
rural and agribusiness
healthcare management
e-business
Converted
0.0
1.0
0
500
1000
1500
2000
count


Last Activity
page visited on website
email opened
others
converted to lead
olark chat conversation
email bounced
email link clicked
form submitted on website
sms sent
Converted
0.0
1.0
0
500
1000
1500
2000
count


Specialization
unstated
business
management
Converted
0.0
1.0
0
500
1000
1500
2000
count

Creating Dummy Variables

Column: 'Lead Origin'

```
dummies = pd.get_dummies(df['Lead Origin'], prefix = 'Lead_Origin_', drop_first = True)
df.drop('Lead Origin', axis = 1, inplace = True)
df = pd.concat([df, dummies], axis = 1)
```

Column: 'Lead Source'

```
dummies = pd.get_dummies(df['Lead Source'], prefix = 'Lead_Source_', drop_first = True)
df.drop('Lead Source', axis = 1, inplace = True)
df = pd.concat([df, dummies], axis = 1)
```

Lead_Origin__landing page submission	Lead_Origin__lead add form	Lead_Origin__lead import	Lead_Source__google	Lead_Source__olark chat
0	0	0	0	1
0	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	1	0

Train/Test Split and Feature Scaling

We split our dataset into train and test set with 70/30 ratio

After that, we scaled all the numerical features using the rows in the training set.

```
df_train, df_test = train_test_split(df, train_size = 0.70, test_size = 0.30)
```

```
df_train.shape
```

```
(6246, 36)
```

```
df_test.shape
```

```
(2678, 36)
```

```
numerical_columns
```

```
['TotalVisits', 'Total Time Spent on Website', 'Page Views Per Visit']
```

```
# Applying Feature Scaling to numerical variables
scaler = StandardScaler()

df_train[numerical_columns] = scaler.fit_transform(df_train[numerical_columns])
```

Dropping features using RFE, p-value and variance inflation factor

Using Recursive Feature Elimination for Feature Selection

```
rfe = RFE(estimator=lm, n_features_to_select=15)
rfe = rfe.fit(X_train, y_train)
```

We used statsmodels to build our Logistic Regression model where we used p-value and variance inflation factor to eliminate unnecessary features

Selected Features

	0	1	2
1	Total Time Spent on Website	True	1
4	Lead_Origin__landing page submission	True	1
5	Lead_Origin__lead add form	True	1
6	Lead_Origin__lead import	True	1
8	Lead_Source__olark chat	True	1
11	Lead_Source__reference	True	1
13	Lead_Source__welingak website	True	1
14	Last_Activity__email bounced	True	1
18	Last_Activity__olark chat conversation	True	1
21	Last_Activity__sms sent	True	1
23	Specialization__unstated	True	1
24	What_occupation__housewife	True	1
28	What_occupation__working professional	True	1
32	Last_Notable_Activity__others	True	1
34	Last_Notable_Activity__sms sent	True	1

Building the final Logistic Regression Model

```
lr4 = build(X_train_rfe, y_train)
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:              Converted   No. Observations:                 6246
Model:                            GLM   Df Residuals:                     6233
Model Family:                Binomial   Df Model:                           12
Link Function:                  Logit   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -2626.9
Date:                Mon, 22 May 2023   Deviance:                       5253.8
Time:                        16:43:53   Pearson chi2:                 6.30e+03
No. Iterations:                     7   Pseudo R-squ. (CS):             0.3882
Covariance Type:            nonrobust
===============================================================================================================
                                            coef    std err          z      P>|z|      [0.025      0.975]
---------------------------------------------------------------------------------------------------------------
const                                    -0.4387      0.124     -3.538      0.000      -0.682      -0.196
Total Time Spent on Website               1.1131      0.040     27.692      0.000       1.034       1.192
Lead_Origin__landing page submission     -1.0646      0.129     -8.282      0.000      -1.317      -0.813
Lead_Origin__lead import                  1.2644      0.482      2.622      0.009       0.319       2.209
Lead_Source__olark chat                   1.1864      0.124      9.604      0.000       0.944       1.429
Lead_Source__reference                    3.3094      0.239     13.874      0.000       2.842       3.777
Lead_Source__welingak website             5.8156      0.730      7.968      0.000       4.385       7.246
Last_Activity__email bounced             -2.4092      0.377     -6.387      0.000      -3.148      -1.670
Last_Activity__olark chat conversation   -1.4864      0.169     -8.790      0.000      -1.818      -1.155
Specialization__unstated                 -1.1158      0.125     -8.954      0.000      -1.360      -0.872
What_occupation__working professional     2.6397      0.194     13.622      0.000       2.260       3.019
Last_Notable_Activity__others             1.1011      0.272      4.045      0.000       0.568       1.635
Last_Notable_Activity__sms sent           1.5909      0.080     19.994      0.000       1.435       1.747
===============================================================================================================
```

Accuracy
80.3%

Sensitivity
81.5%

Specificity
79.5%

Recommendations

Take advantage of the leads coming from these:

- Lead Origin: add form
- Lead Source: reference/welingak website
- Total Time Spent on website: Above 500
- Last Activity: sms sent
- Occupation: working professional

Try to improve upon the lead conversion rate from these:

- Lead Origin: api/landing page submission
- Lead Source: olark chat/organic search/direct traffic/google
- Total Time Spent on website: Below 500
- Last Activity: email opened
- Occupation: unemployed