Lead Scoring Case Study

By

Aniket Muley

Yashashri Jounjal

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 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

- We have been provided with a leads dataset from the past with around 9000 data points.
- This dataset consists of various attributes such as Lead Source, Total Time Spent on Website, Total Visits, Last Activity, etc. which may or may not be useful in ultimately deciding whether a lead will be converted or not.
- The target variable, in this case, is the column ‘Converted’ **which tells whether a past lead was converted or not wherein 1 means it was converted and 0 means it wasn’t converted**.

Data importing and cleaning

- We import the data and perform data cleaning operation on them.
- We drop the following columns as this columns have same value which was not much impact on the data.
  - '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',
  - 'Do Not Email', 'A free copy of Mastering The Interview', 'City', 'Country', 'Prospect ID‘

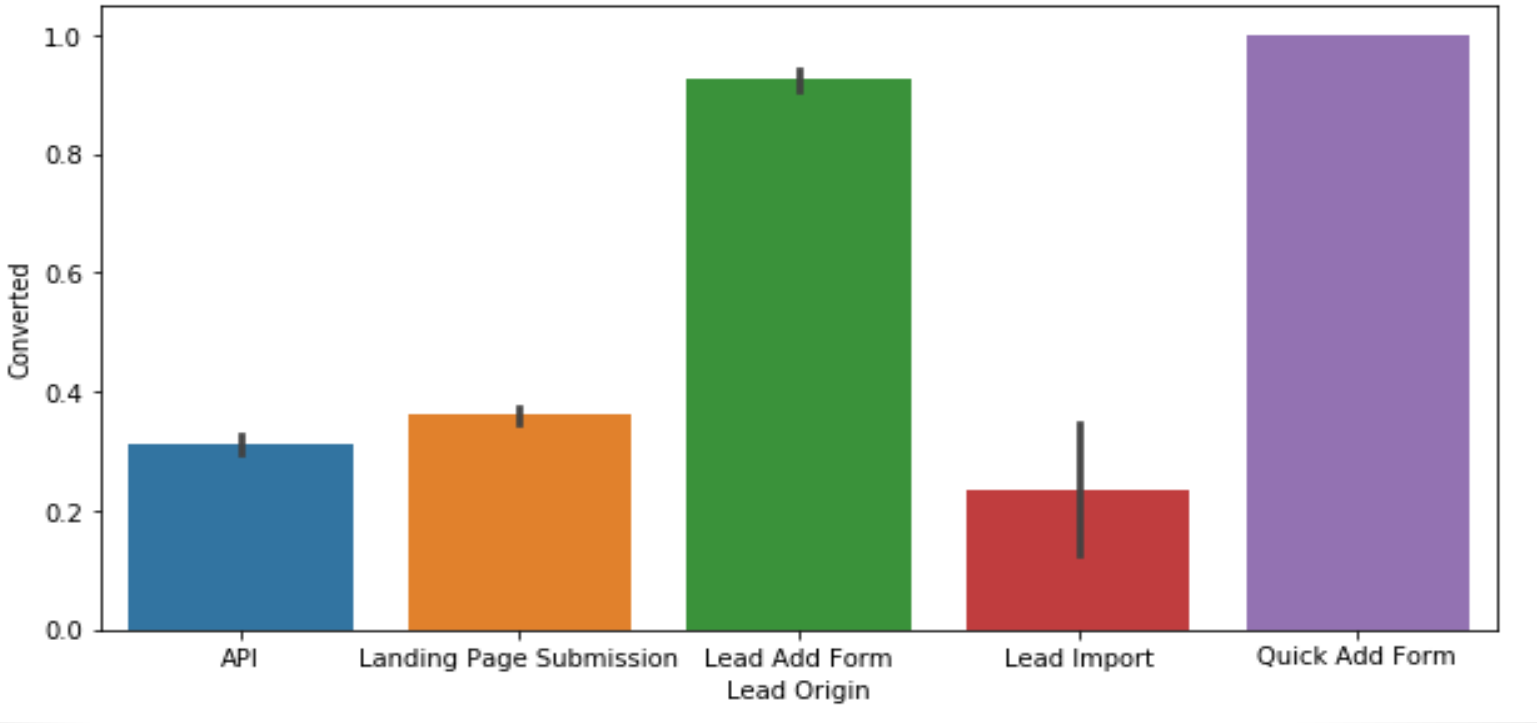
Data Cleaning and treating missing values

- Since there are more percentage of null values in these columns it won't make any sense in having them in the dataframe
- So we remove all those columns which are having null values more than 45%

  - **And also we fill the NaN values with zero as this will not much made any difference because of huge size of data , it'll not affect that much**

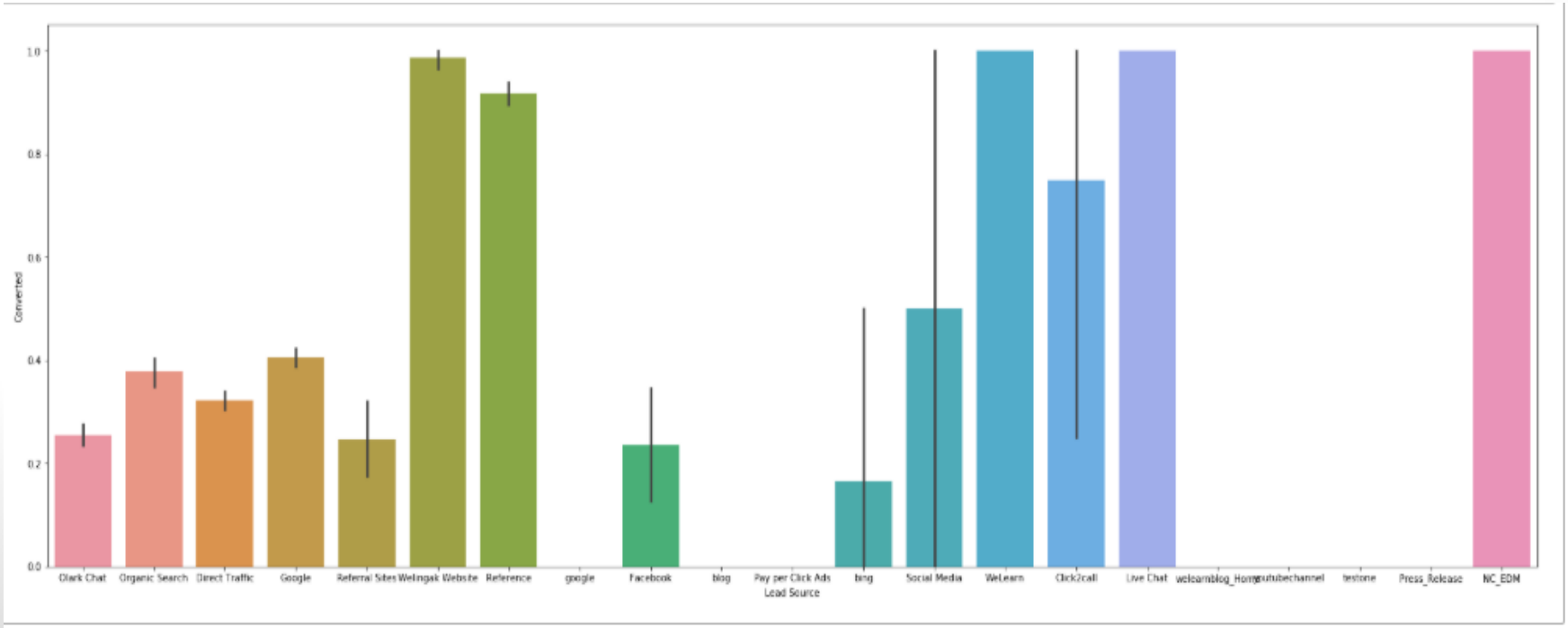
Performing univariate and Bivariate Analysis

- **Comparing the Lead Origin with the Converted**
- **We can say that Quick add form has great effect on converted on leads means it can convert more to Hot Leads**



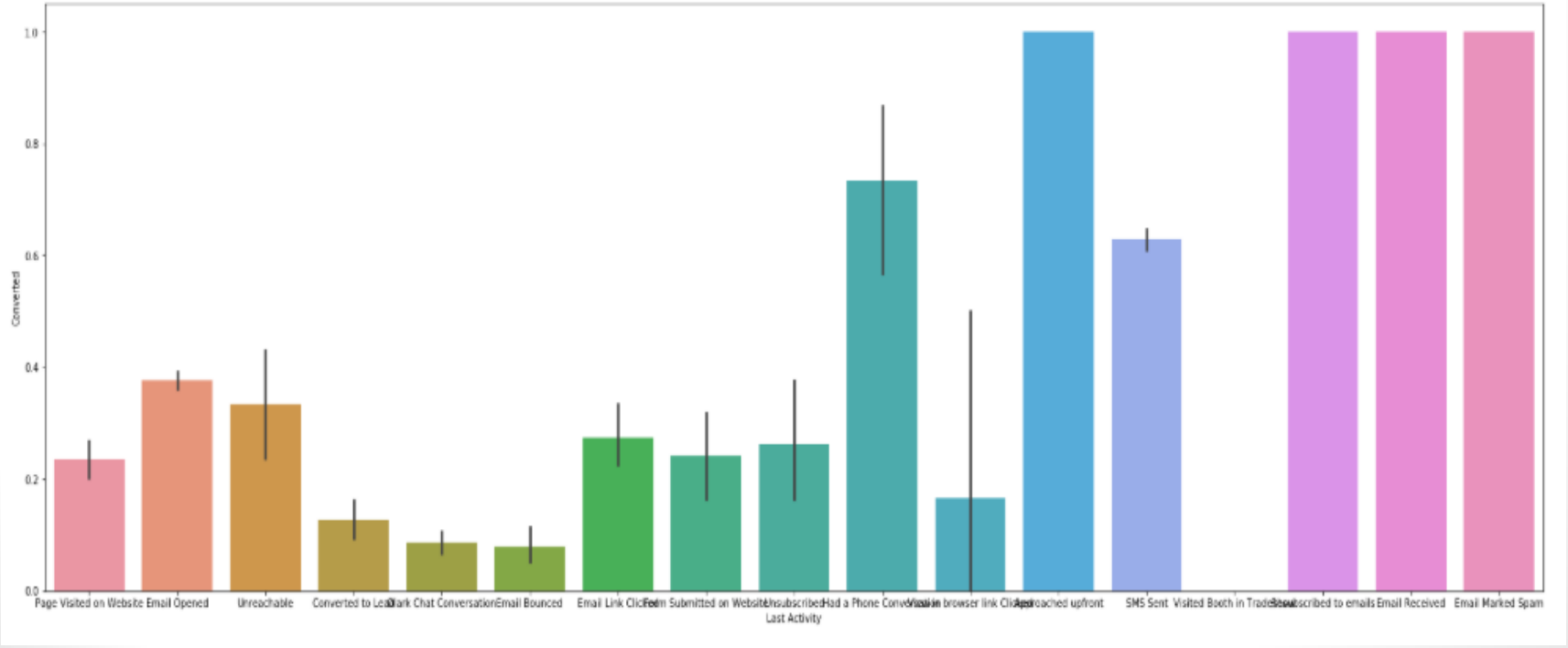
Comparing the Lead Source with the Converted

- Welingak Website,Livechat,WeLearn ,References are the Lead source which only says source of accessing the content of that company



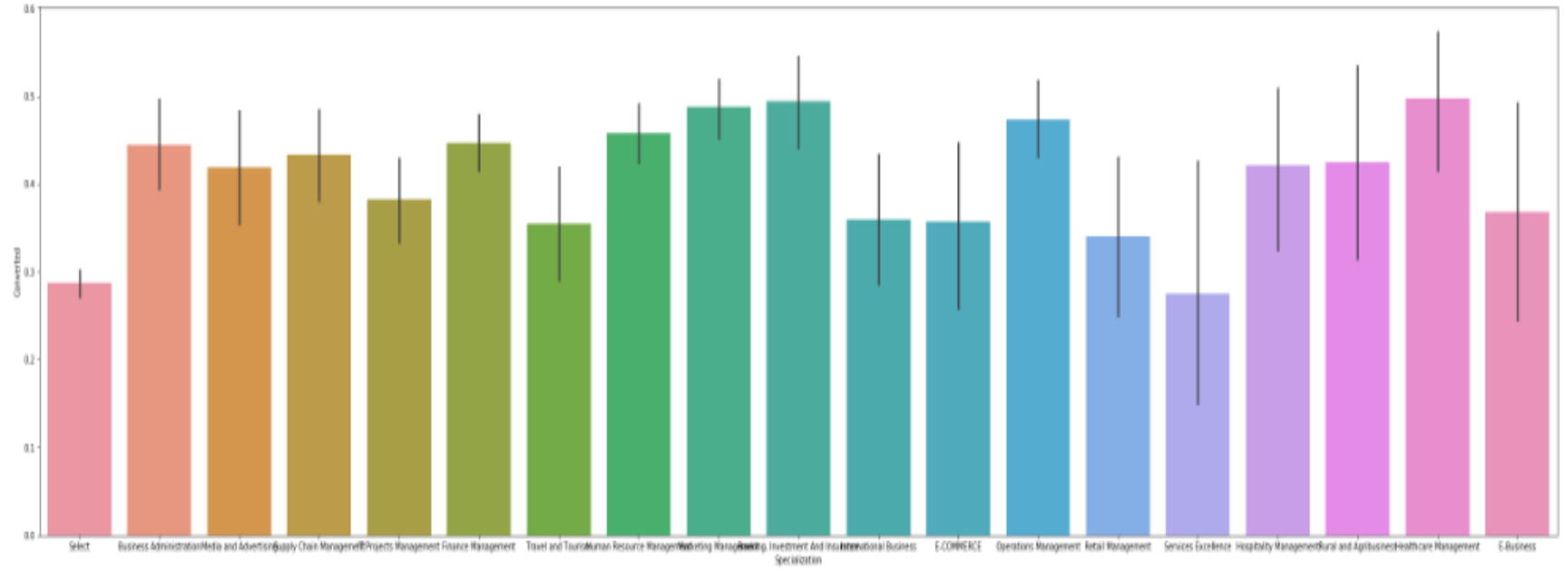
Comparing the Last Activity with the Converted

- Subscribed to Email, Email received , Marked as Spam as some impact on converted ,it can tell that if the Lead gonna convert it or not, if they marked as spam , definitely they're not interested.



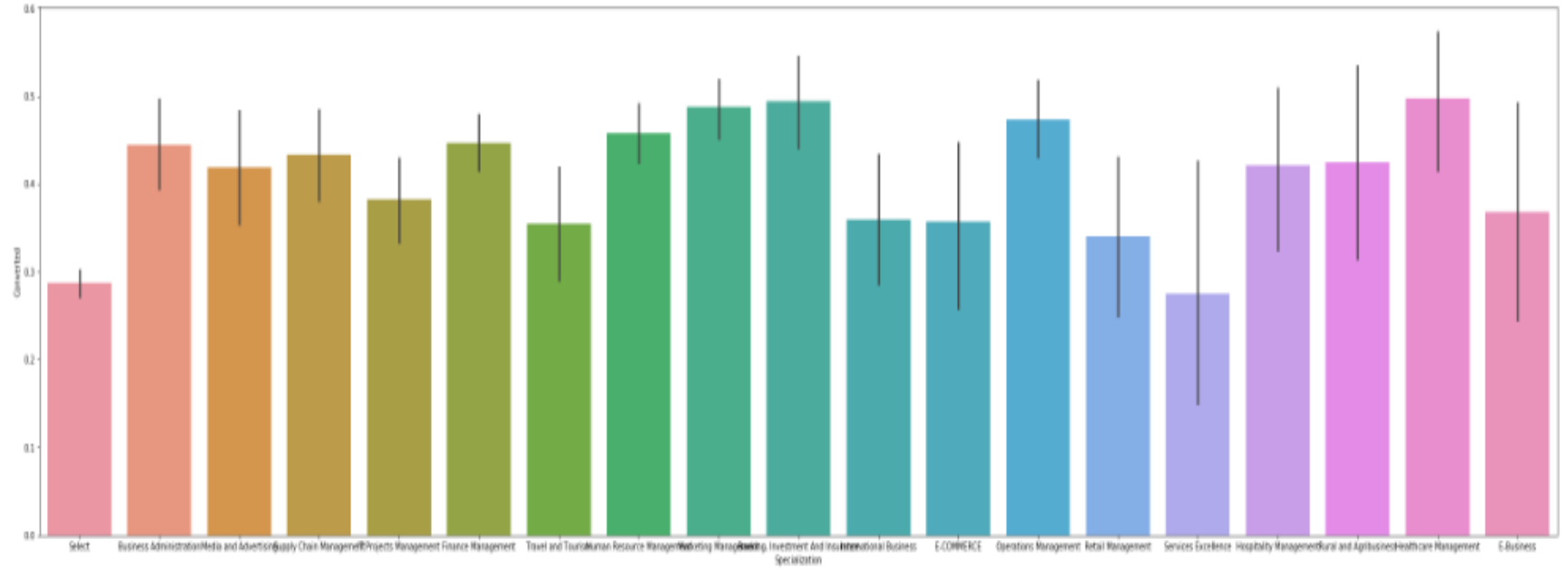
Comparing the Last Activity with the Specialization

- Rural and agribusiness , Healthcare and Management like this specialization were less for converted rate as this were not selected while filing form.



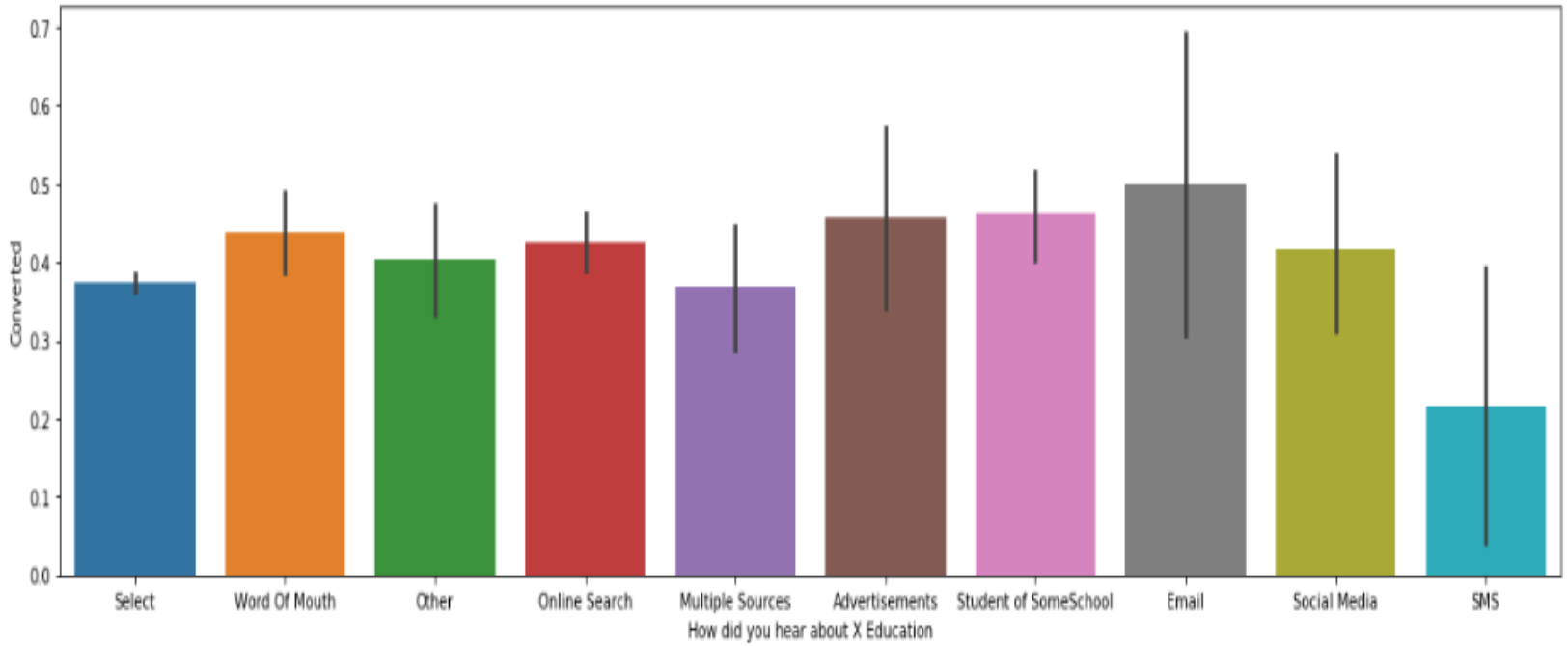
Comparing the Last Activity with the Specialization

- Rural and agribusiness , Healthcare and Management like this specialization were less for converted rate as this were not selected while filing form.



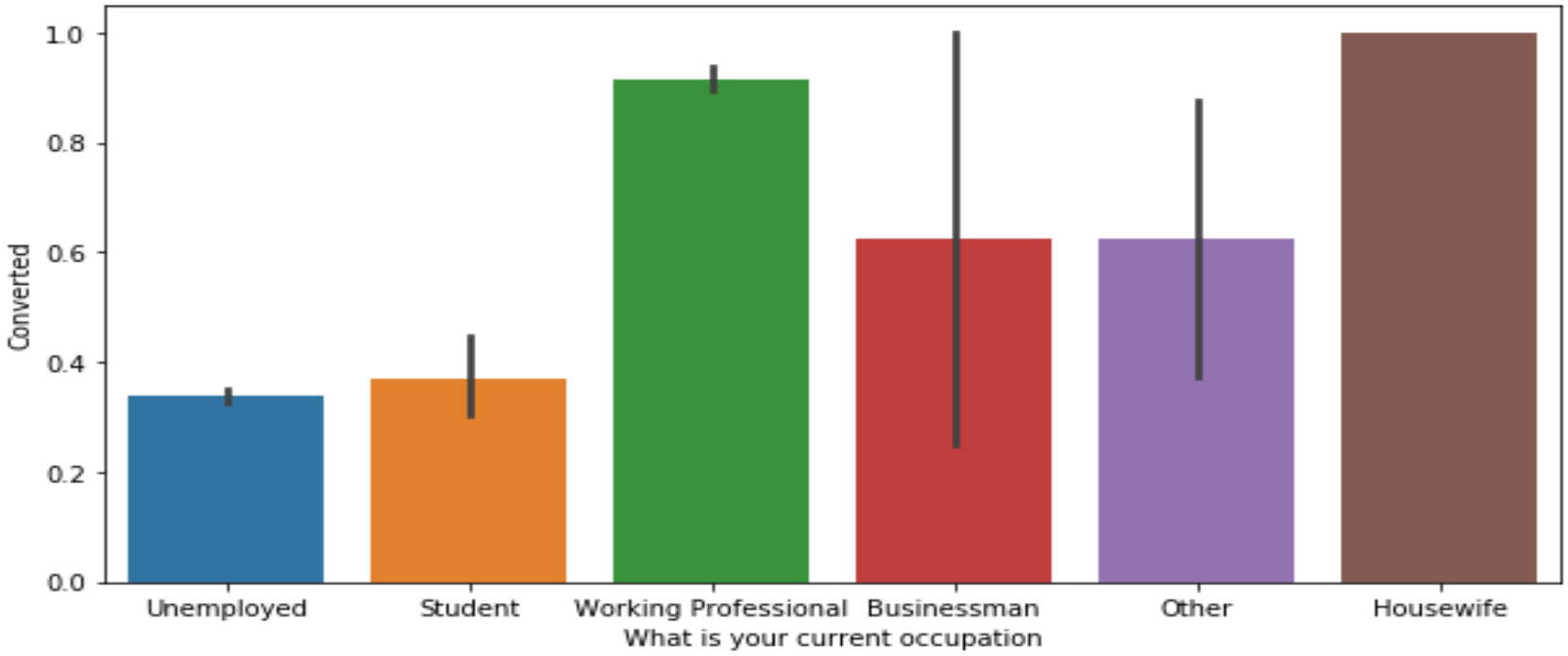
Comparing the How did you hear about X Education with the Converted

- From the visualisation , through email and Student of some school they get hearing about the X Education , so we can focus on that fileds more to get leads.



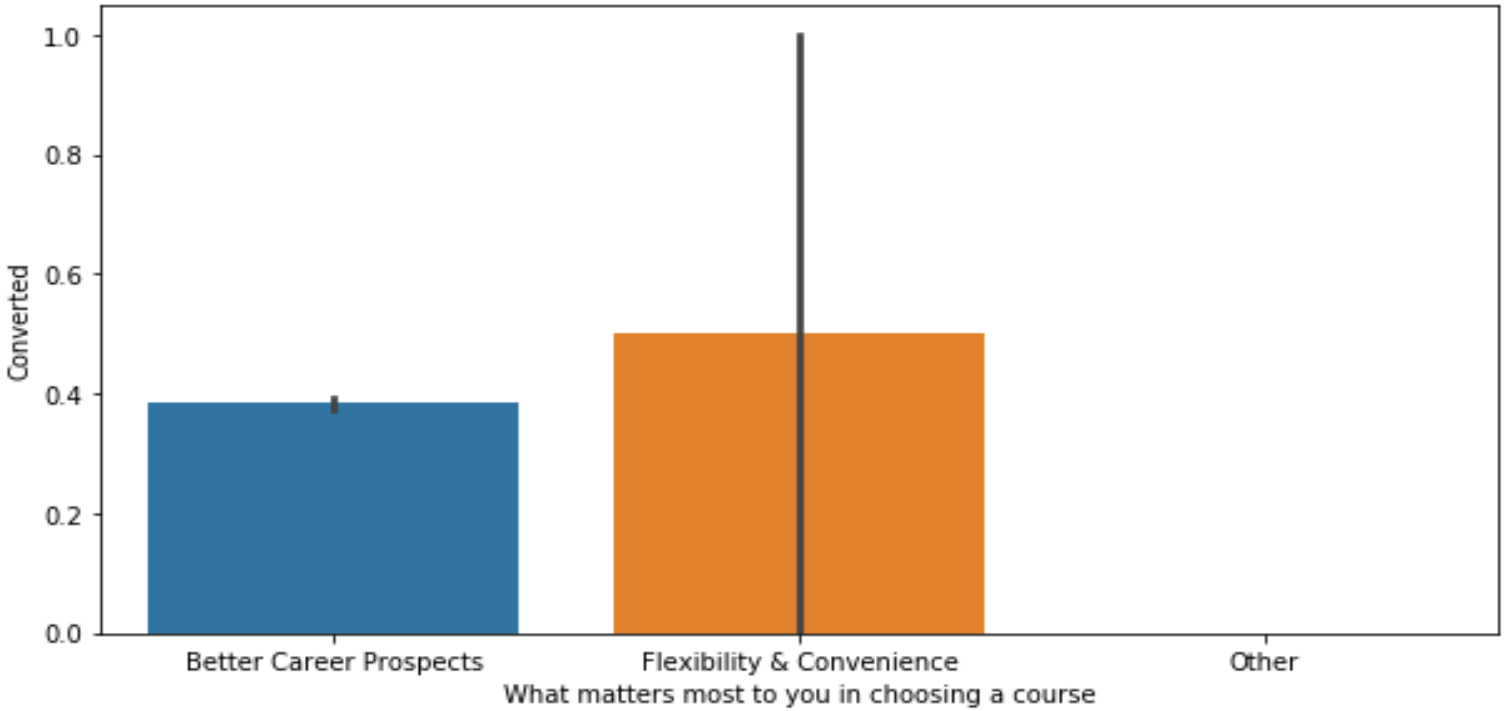
Comparing the What is your current occupation with the Converted

- From the visualisation , we get the idea of that the housewife and working professional are are the majority converted leads that they have greater impact



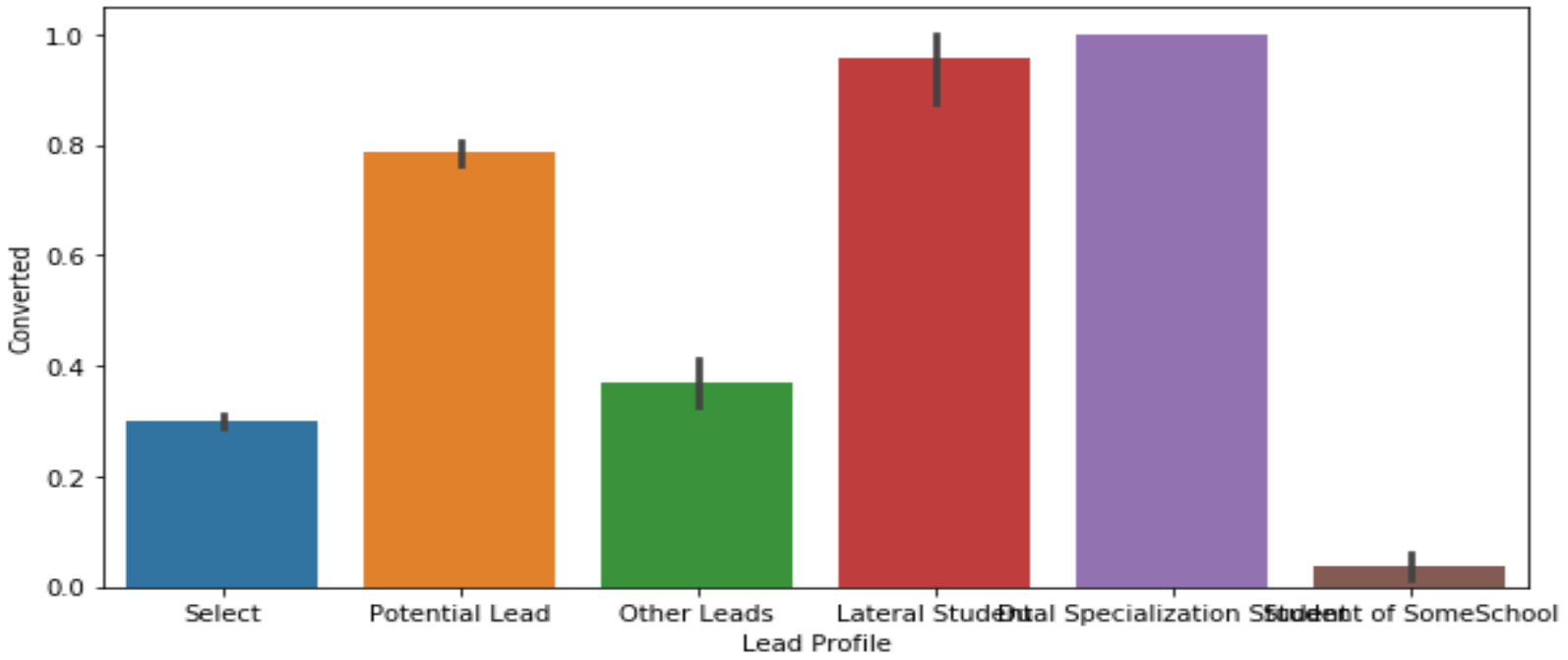
Comparing the What matters most to you in choosing a course with the Converted

- From the visualisation , we get that the course should be flexible and convenience then there is converted leads are hot.



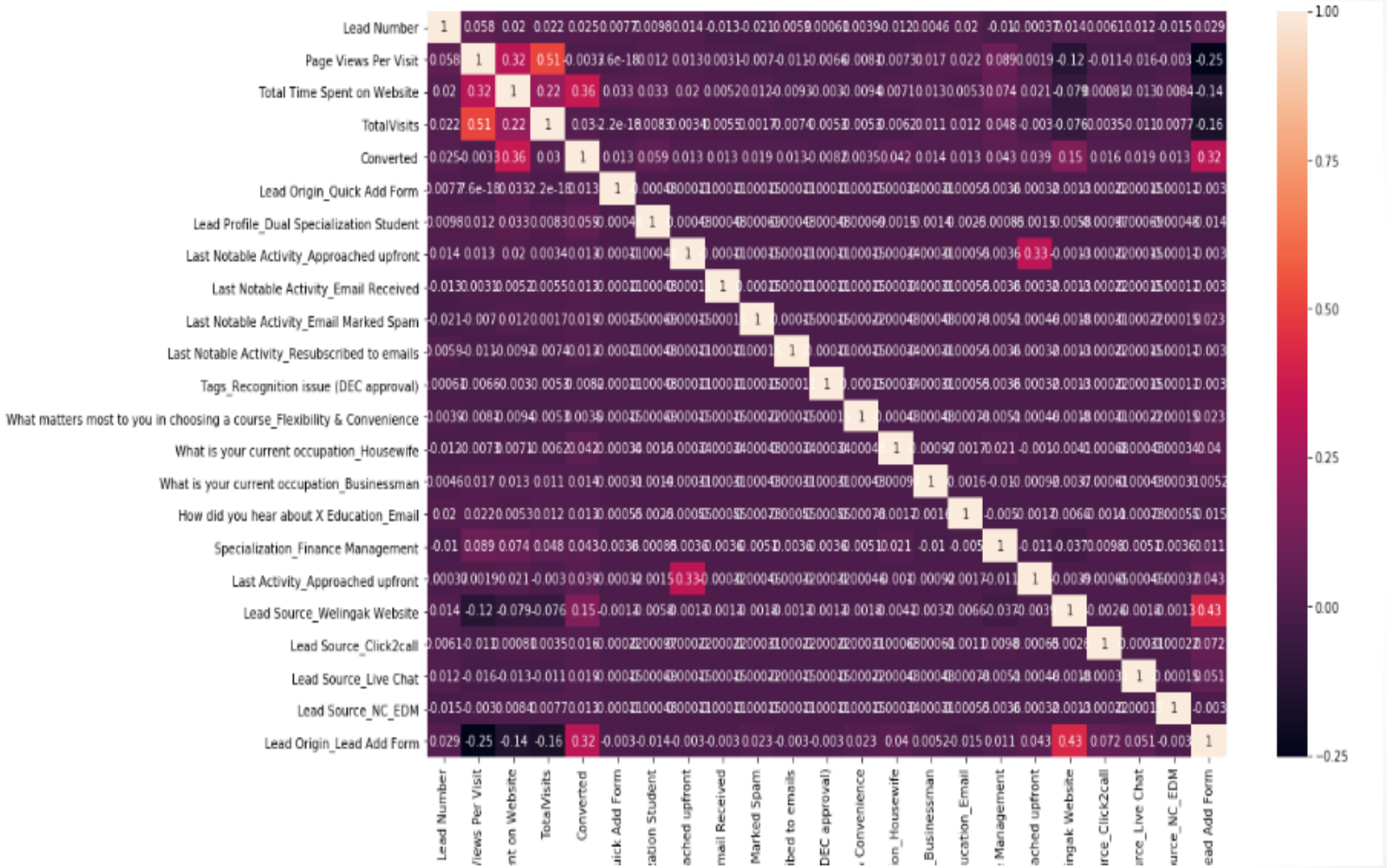
Comparing the Lead Profile with the Converted

- A lead level assigned to each customer based on their profile therefor the Dual specialization Student and lateral student this lead level is assigned student are Hot leads that there is converted.



- After this Analysis we selected this fields as most affecting and then we created the Dummies from below variable.
- 'Lead Number', 'Page Views Per Visit', 'Total Time Spent on Website', 'TotalVisits','Converted‘, 'Lead Origin_Quick Add Form', 'Lead Profile_Dual Specialization Student', 'Last Notable Activity_Approached upfront', 'Last Notable Activity_Email Received', 'Last Notable Activity_Email Marked Spam', 'Last Notable Activity_Resubscribed to emails‘ ,'Tags_Recognition issue (DEC approval)','What matters most to you in choosing a course_Flexibility & Convenience','What is your current occupation_Housewife‘ , 'What is your current occupation_Businessman‘, 'How did you hear about X Education_Email', 'Specialization_Banking','Specialization_Finance Management', 'Investment And Insurance','Last Activity_Approached upfront', 'Lead Source_Welingak Website','Lead Source_Click2call','Lead Source_Live Chat','Lead Source_NC_EDM','Lead Origin_Lead Add Form'

Correlation Matrix



Splitting the data into training and test data

- **Generalized linear model (GLM)** is the basis of many machine-learning algorithms. GLM with first-order variables is basically linear regression, and can be analytically solved (meaning there is a formula that you can use to solve the GLM problems). If you add variables of higher order (order 2 or more), you can fit the data with non-linear trend.

- Now we are running the logistic model for dataset and the use the REF which is Recursive Feature Elimination (RFE) is based on the idea to repeatedly construct a model and choose either the best or worst performing feature, setting the feature aside and then repeating the process with the rest of the features. This process is applied until all features in the dataset are exhausted. The goal of RFE is to select features by recursively considering smaller and smaller sets of features.

Feature Selection Using RFE

- We are running the RFE with 14 variables as you can see below:

	coef	std err	z	P>|z|	[0.025	0.975]
const	-9.4938	0.915	-10.381	0.000	-11.286	-7.701
Lead Number	1.157e-05	1.46e-06	7.935	0.000	8.71e-06	1.44e-05
Lead Origin	0.4578	0.070	6.522	0.000	0.320	0.595
Lead Source	0.1521	0.025	6.119	0.000	0.103	0.201
TotalVisits	0.0191	0.008	2.405	0.016	0.004	0.035
Total Time Spent on Website	0.0016	6.42e-05	25.602	0.000	0.002	0.002
Page Views Per Visit	-0.2363	0.021	-11.049	0.000	-0.278	-0.194
Last Activity	0.0324	0.010	3.377	0.001	0.014	0.051
Specialization	-0.0014	0.008	-0.183	0.855	-0.017	0.014
How did you hear about X Education	-0.0265	0.020	-1.346	0.178	-0.065	0.012
What is your current occupation	1.1293	0.072	15.773	0.000	0.989	1.270
What matters most to you in choosing a course	-0.5068	1.546	-0.328	0.743	-3.538	2.524
Tags	-0.0411	0.009	-4.512	0.000	-0.059	-0.023
Lead Profile	0.1844	0.034	5.502	0.000	0.119	0.250
Last Notable Activity	0.1796	0.013	14.182	0.000	0.155	0.204

Variables selected by RFE and then again logistic regression model fitting

- 'Lead Number', 'Lead Origin', 'Lead Source',
- 'TotalVisits', 'Total Time Spent on Website',
- 'Page Views Per Visit', 'Last Activity',
- 'Specialization', 'How did you hear about X Education',
- 'What is your current occupation',
- 'What matters most to you in choosing a course', 'Tags',
- 'Lead Profile', 'Last Notable Activity'

Making the Prediction of test set results and calculating the Accuracy

- Creating the new column 'predicted' with 1 if Converted_Prob>0.5 else 0 ,so the converted candicate are 1 and not converted are 0.

	Lead Number	Converted	Converted_Prob	predicted
0	4269	1	0.553222	1
1	2376	1	0.226568	0
2	7766	1	0.258066	0
3	9199	0	0.246344	0
4	4359	1	0.232991	0

Confusion Matrix and final Accuracy is 72%

```
1 # Confusion matrix
2 confusion = metrics.confusion_matrix( y_pred_final.Converted, y_pred_final.predicted )
3 confusion
```

```
array([[1457,  220],
       [ 553,  542]], dtype=int64)
```

```
1 #Let's check the overall accuracy.
2 metrics.accuracy_score(y_pred_final.Converted, y_pred_final.predicted)
```

```
0.7211399711399712
```

Running Your First Training Model

- By running the Training data set we get the accuracy as 72.11%

```
1  from sklearn.metrics import confusion_matrix
2  confusion_matrix(y_test,prediction)
```

```
array([[1457,  220],
       [ 553,  542]], dtype=int64)
```

```
1  from sklearn.metrics import accuracy_score
2  accuracy_score(y_test,prediction)
```

```
0.7211399711399712
```