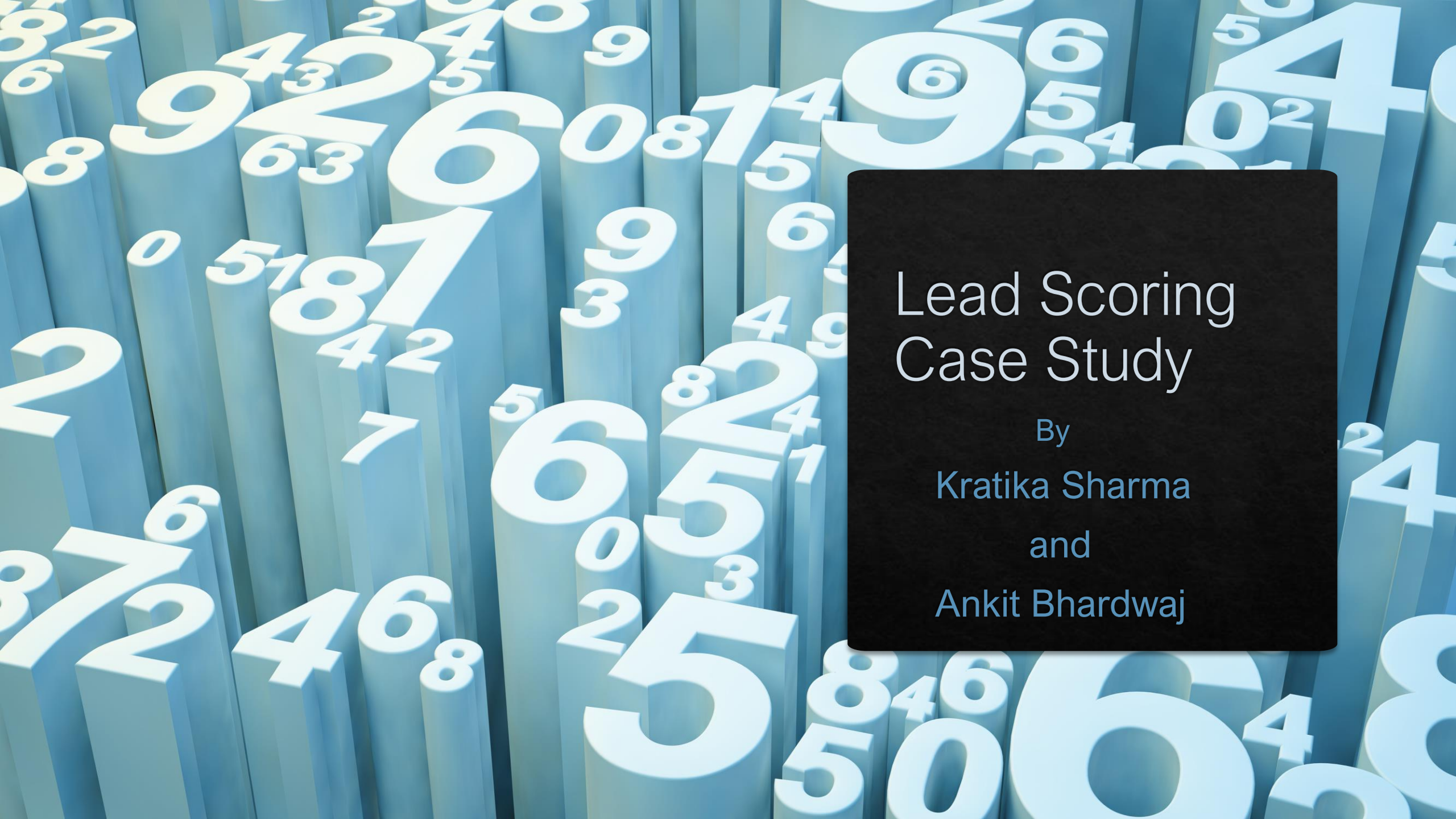


Lead Scoring Case Study

By

Kratika Sharma

and

Ankit Bhardwaj

Problem Statement

- X Education company sells online courses to industry professionals.
- X Education despite getting many leads, its lead conversion rate is poor, for every 100 leads in a day, only 30 are converted.

Business Goals

To identify the most promising leads that are most likely to convert into paying customers

To build a model for the company by assigning a lead score to each of the customer according to their chances of conversion with higher scores indicating greater chance of conversion.

Steps

- Reading Data
- Data Cleaning
- Exploratory Data Analysis
- Creating Dummy Variables
- Train and Test split
- Model Building
- Model Evaluation
- Conclusions

Reading and Inspecting Data

- Data inspection – Checking shape, info, dtypes, duplicates etc
- Data cleaning and preparation
- Replacing ‘seven’ values across various columns by np.nan
- Checking null value percentages and dropping columns with more than 40% missing values.
- Dropping all skewed categorical columns
- Performing various imputation techniques such as mean, median and mode for columns with lesser missing values
- Check the percentage of rows retained in data cleaning process.

Exploratory Data Analysis

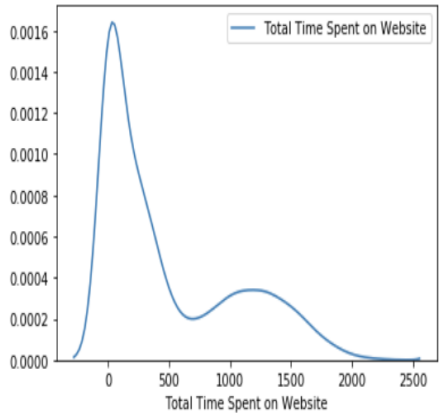
Univariate Analysis

EDA

Uni-Cont

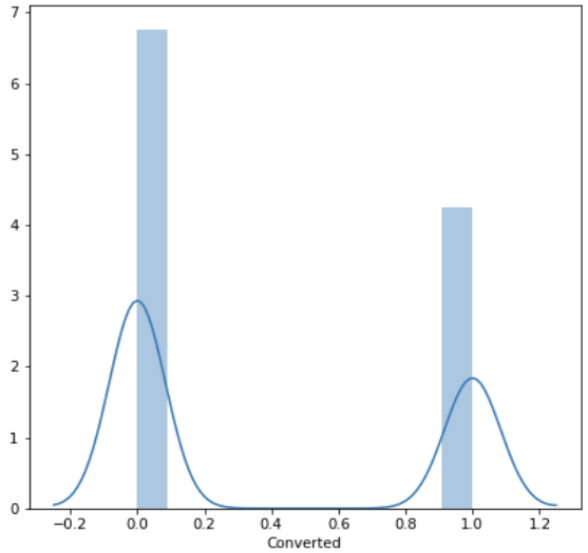
In [421]:
```
def unicont(columnName):
    sns.distplot(df_lead[columnName],hist=False,label = columnName)
```

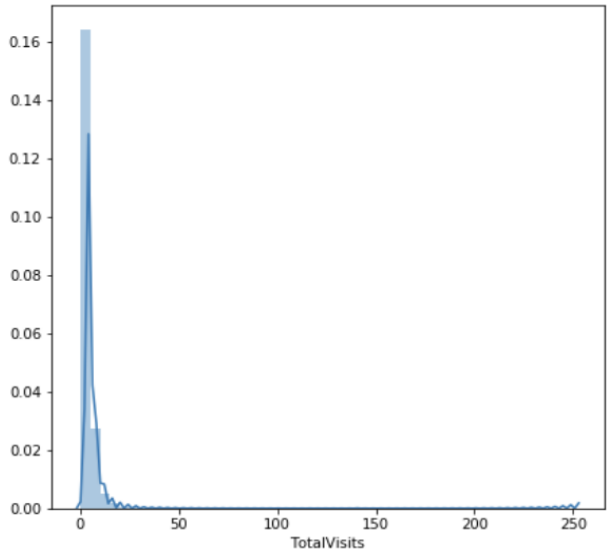
In [422]:
```
unicont("Total Time Spent on Website")
```



Bivariate Analysis

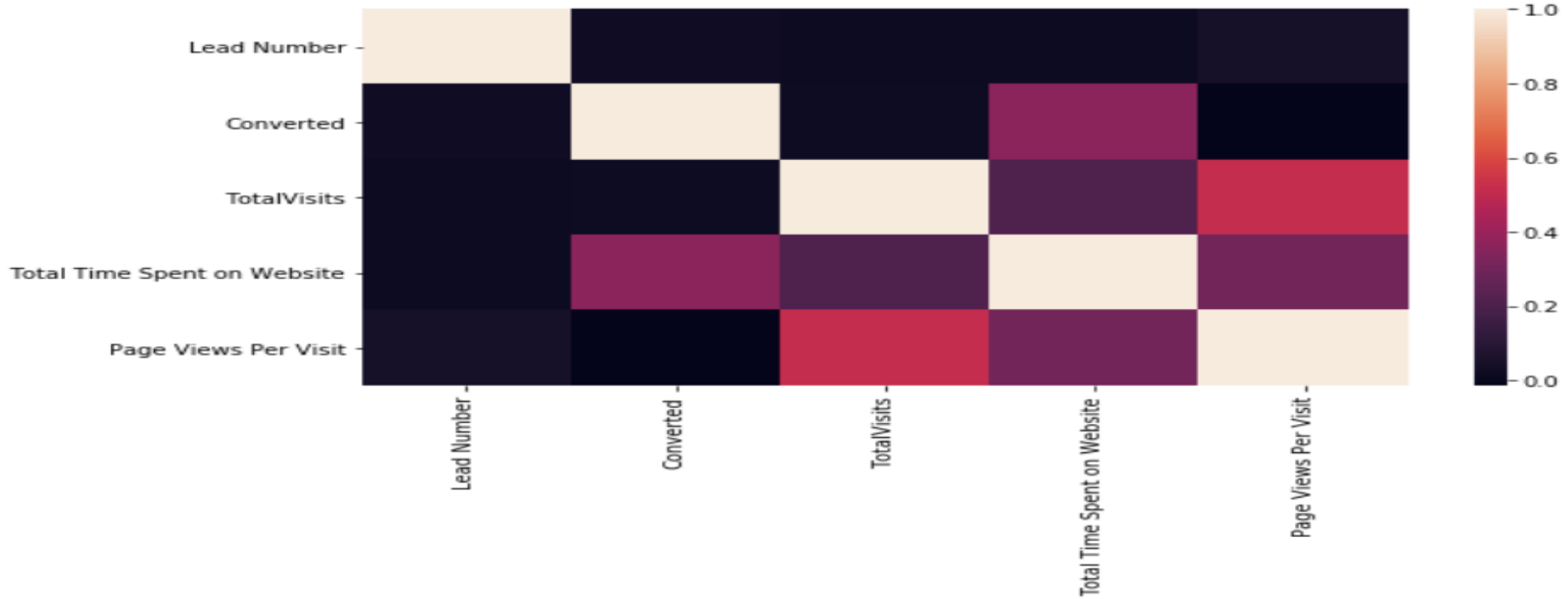
In [425]:
```
varlist =  ['Converted','TotalVisits']
plt.figure(figsize = (15,15))
f = df_lead[varlist]
for i in enumerate(f):
    plt.subplot(2,2,i[0]+1)
    sns.distplot(df_lead[i[1]])
```





EDA (Heatmap)

```
In [426]: plt.figure(figsize=(10,5))
          sns.heatmap(df_lead.corr())
          plt.show()
```



Looking at above corelations we can say that there is a positive corelation in converted and Total Visits

Data Preparation

- Creating dummy variables for all categorical columns
- Perform train-test split
- Perform scaling

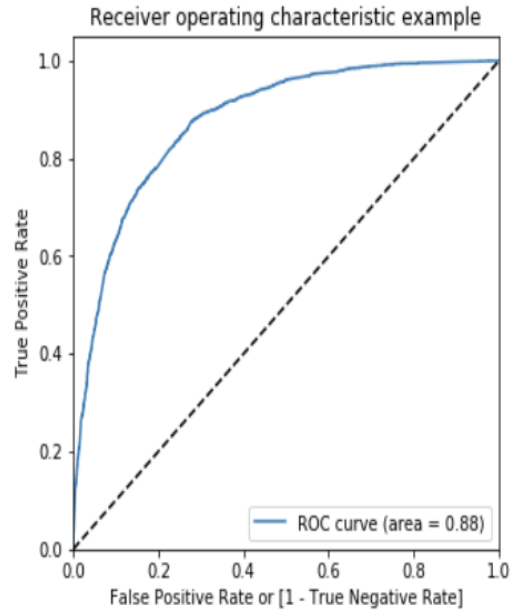
Model Building

- Using RFE to perform variable selection
- Building a Logistic Regression model with good sensitivity
- Manual selection by checking and removing variable with values p-value>0.05 and VIF>5
- Finding the optimal probability cutoff
- Predictions on test data set

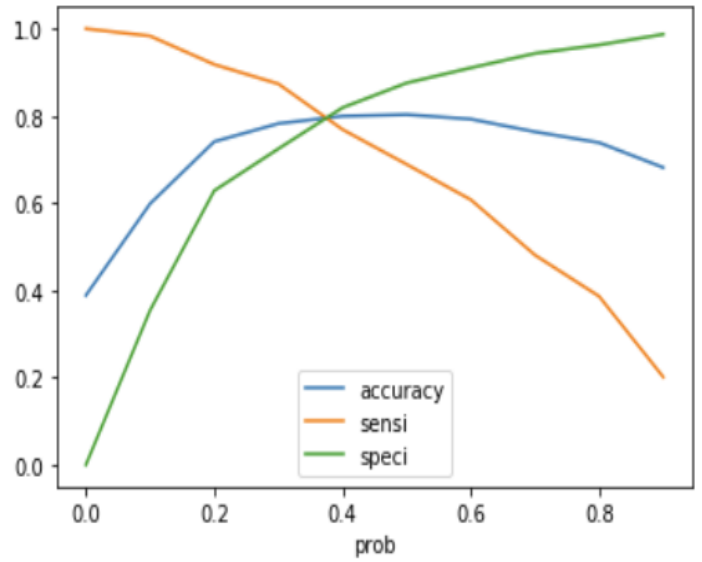
ROC Curve

- Finding Optimal Cutoff point
- Optimal probability is that where we get balanced sensitivity and specificity.

```
In [504]: draw_roc(y_train_pred_final.Converted, y_train_pred_final.Conversion_Prob)
```




```
In [508]: cutoff_df.plot.line(x='prob', y=['accuracy','sensi','speci'])
plt.show()
```



Conclusion

- The value of Precision and Recall on the test data set is 73 % and 76.5 % respectively.
- The variables that affect the conversion of a visitor are
- Total time spent on website
- Lead Origin
- Lead Source

THE END