Data Preprocessing, Feature Selection, and Model Optimization

Steps for Data Exploration, Feature Selection, and Model Optimization

Import Necessary Libraries

- Key Points:
  - pandas, numpy
  - seaborn, matplotlib
  - sklearn.metrics (accuracy_score)
  - optbinning (BinningProcess)

Data Exploration

- Key Points:
  - Check for duplicates and missing values
  - Identify and manage outliers
  - Example code for missing values, duplicates, and outliers visualization

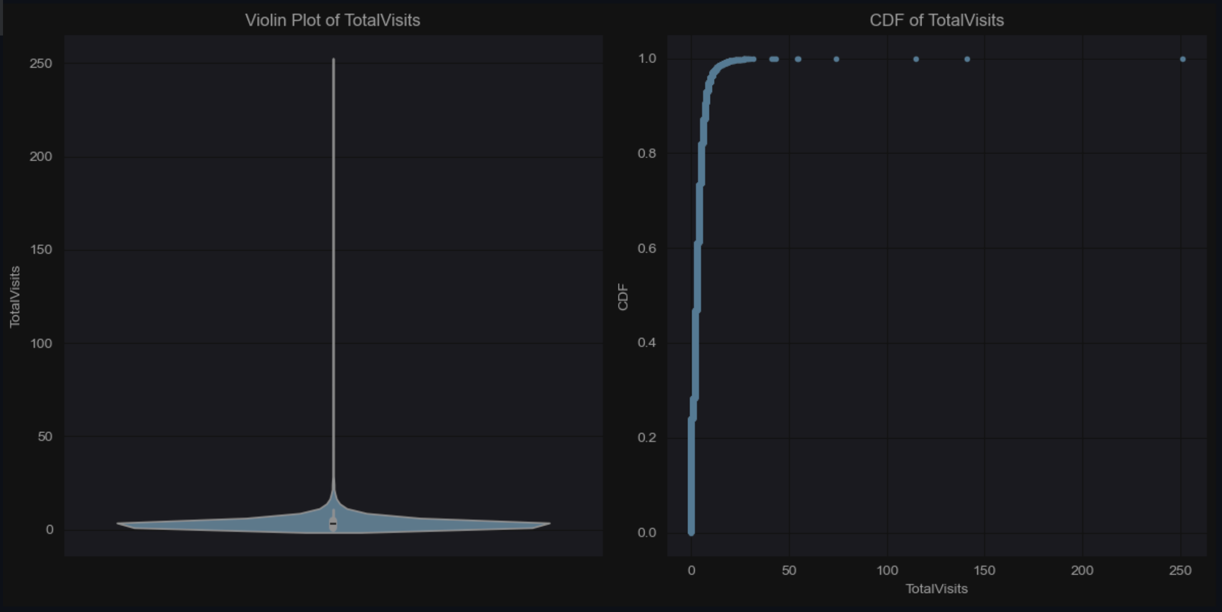
```
missing_values_percentage = round(100 * (data.isna().sum() / len(data)), 2)
missing_values_percentage_sorted = missing_values_percentage.sort_values(ascending=False)
missing_values_percentage_sorted
```

✓ [9] 13ms

1-10 Length: 35, dtype: float64

	<unnamed>
Lead Quality	51.59
Asymmetrique Activity Index	45.65
Asymmetrique Profile Score	45.65
Asymmetrique Profile Index	45.65
Asymmetrique Activity Score	45.65
Tags	36.29
Lead Profile	29.32
What matters most to you in choosing a course	29.32
What is your current occupation	29.11
Country	26.63

```
TotalVisits
25th percentile: 1.0
50th percentile: 3.0
75th percentile: 5.0
90th percentile: 7.0
99th percentile: 17.0
```



The pictures demonstrate that their lot of nan value in each columns. The most interesting that there are outliers for Total Visit columns

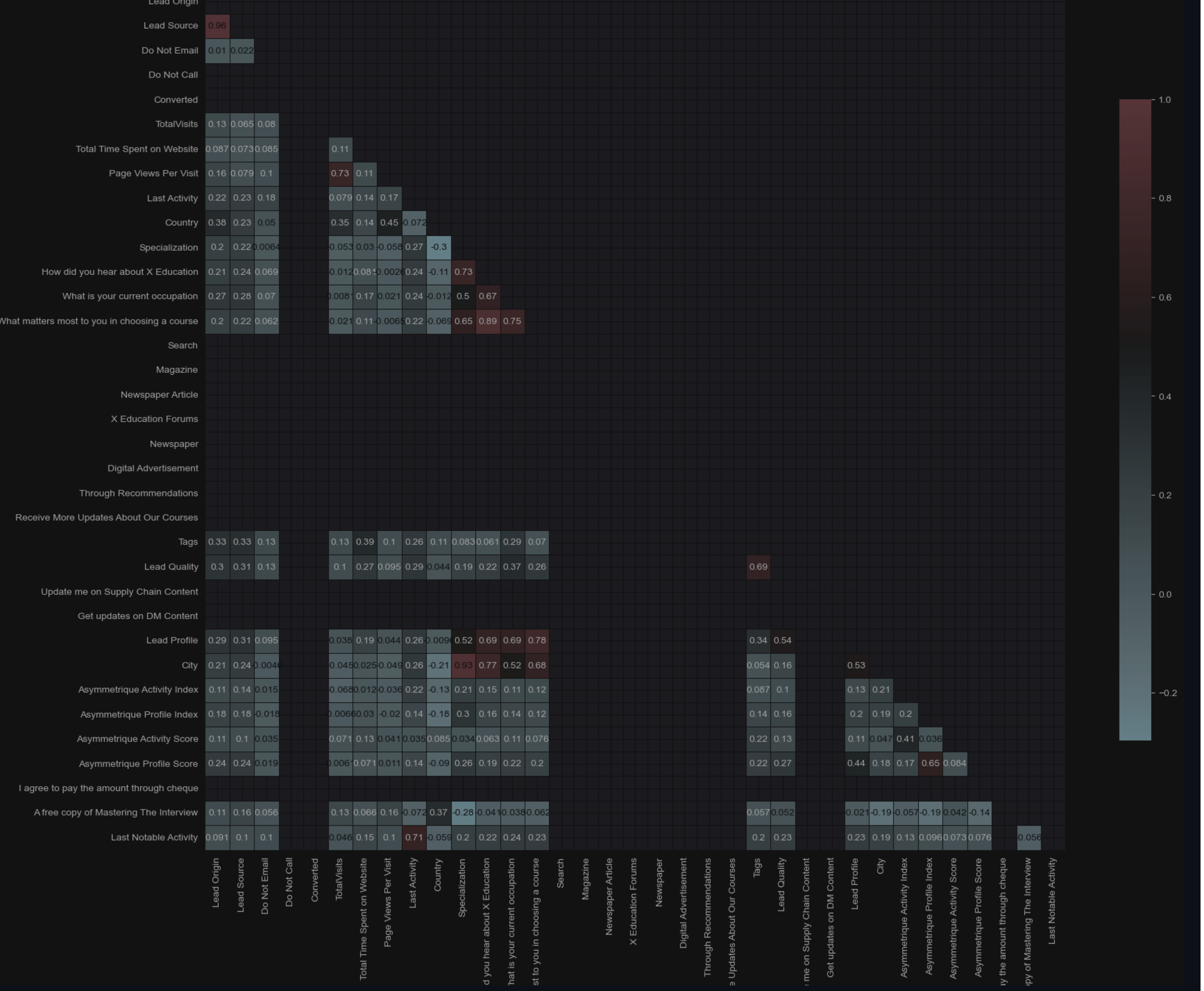
Feature Selection Using Correlation and Information Value (IV)

- Key Points:
  - Pearson correlation to check multicollinearity
  - Heatmap visualization
  - Remove highly correlated features with low IV values

		Bin	Count	Count (%)	Non-event	Event	Event rate	WoE	IV	JS
Lead Origin	0	[3 0]	2911	0.3938041	2020	891	0.3060804	0.3518888	0.04641371	0.00577196
Lead Origin	1	[1]	3903	0.5280032	2484	1419	0.3635665	0.0932982	0.00454416	0.00056781
Lead Origin	2	[2 4]	578	0.0781926	39	539	0.932526	-3.0927735	0.55856897	0.05087976
Lead Origin	Totals		7392	1	4543	2849	0.3854167		0.60952684	0.05721954
Lead Source	0	[18 9 19 17	1554	0.2102273	1176	378	0.2432432	0.6683604	0.08433482	0.01034992
Lead Source	1	[1]	2049	0.2771916	1377	672	0.3279649	0.2507846	0.01686061	0.00210207
Lead Source	2	[7]	913	0.1235119	569	344	0.3767798	0.0366192	0.00016492	2.06E-05
Lead Source	3	[3]	2295	0.3104708	1381	914	0.3982571	-0.0538869	0.00090693	0.00011335
Lead Source	4	[21 10 14 13	581	0.0785985	40	541	0.9311532	-3.0711594	0.5561453	0.05083947
Lead Source	Totals		7392	1	4543	2849	0.3854167		0.65841257	0.06342542

Column1	variable	iv	unique_bin	top_bin	freq_bin
18	Tags	4.82413	5	[16 26 20 5 18 15 1]	2842
19	Lead Quality	2.008334	5	[5 3]	4664
22	Lead Profile	1.088576	4	[4]	3314
31	Total Time Spent on Website	1.065929	5	[1.50, 416.50)	2860
8	What is your current occupat	1.007207	3	[3 4 0 2]	4694
4	Last Activity	0.845716	4	[3 13 5]	3054
28	Last Notable Activity	0.661166	4	[6 9 1 8]	2931
1	Lead Source	0.658413	5	[3]	2295
0	Lead Origin	0.609527	3	[1]	3903
9	What matters most to you in	0.572587	2	[0 1]	5249
7	How did you hear about X Ec	0.478849	4	[6 1 0]	4122
6	Specialization	0.384737	5	[14 15 17 6 1 11]	2486
33	Asymmetrique Activity Score	0.383068	5	Missing	3355
23	City	0.356867	5	[6 0]	2645
34	Asymmetrique Profile Score	0.182607	5	Missing	3355
2	Do Not Email	0.108354	2	[0]	6794

I exported the dataframe to binning_table.csv and iv.csv to know which variables has the top IV and need to be input to the model



After that, transfor the original input to dataframe, replaced it with WOE value. There are some conditions that need to filter for useful variables:
The correlation must be below 0.7, if it is greater than 0.7 -> eliminate the lower IV variable. Only choose the varibles which has the IV > 0.07

Model Performance Summary

- **Key Points:**
  - **Baseline Logistic Regression:**
    - Accuracy: 0.81
    - AUC: 0.88
  - **Optimized Logistic Regression:**
    - Accuracy: 0.789
    - AUC: 0.940
  - **Optimized XGBoost:**
    - Accuracy: 0.904
    - AUC: 0.966

Threshold Optimization for XGBoost

- **Key Points:**
  - Adjust decision threshold to optimize accuracy
  - Example code for finding the best threshold
  - **Best Threshold:** 0.45
  - **Best Accuracy:** 0.906

I choose the Xgboost for choosing the threshold because the performance of this model seems outstands the other 2 model (Baseline Logistic regression model and Optimzed Logistic regression model)