LEAD SCORING CASE STUDY

Analysis Presented by:
Aditi Marwaha and Smriti Pradhan

PROBLEM STATEMENT

Company X is an Education platform that is facing Low Lead Conversion rate despite a high number of leads generated.

The Conversion rate before the model is 30%, by identifying the "Hot Leads" i.e. leads most likely to convert to paying customers, we have increased the success of the Company X for making the sale.

The Target Lead Conversion is 80%

LEAD GENERATION:

Lead generation is crucial for any business as it helps identify potential customers who are more likely to convert into paying clients. By focusing on high-quality leads, Company X can optimize their sales and marketing efforts, leading to higher revenue and growth.

The dataset has approx. **9000 data points** with various attributes such as '*Lead Source*', '*Total Time Spent on Website*', '*Total Visits*', '*Last Activity*', etc.

LEAD CONVERSION:

The target variable column 'Converted' tells whether a past lead was converted or not wherein 1 means it was converted and 0 means it wasn't converted. This metric is very important for the purpose of the study.

LEAD CLASSIFICATION:

Leads are classified as potential customers through Company X (**Origin Identifier**) including *'API, Landing Page Submission'* through various **lead sources** like *'Google', 'Organic Search', 'Olark Chat'*, etc.

The need for a lead scoring model is evident to identify the most promising leads among the generated leads. The model has assigned a score (0-100) to each lead, enabling the company to prioritize and focus on leads with a higher likelihood of conversion in order to allocate their resources more efficiently and effectively, ultimately improving the lead conversion rate.

OBJECTIVE

The lead scoring model built using a machine learning algorithm (logistic regression) analyzed the historical data of converted leads and identify the most important attributes that contribute to conversion. To increase the lead score leading to the higher the conversion chance, the following steps have been taken:

| DATA CLEANING	EDA	DATA PREPARATION
BUILDING THE MODEL	EVALUATING THE MODEL	MAKING PREDICTIONS ON THE DATASET

DATA CLEANING

Data cleaning is an essential step in building an accurate lead scoring model. It involves identifying and correcting errors, inconsistencies, and missing values in the dataset.

Python libraries like **Pandas**, **Numpy**, and **Matplotlib** are used for data cleaning. Pandas is used for data manipulation and cleaning, Numpy is used for numerical operations, and Matplotlib is used for data visualization. These libraries provide various functions and methods that help in identifying and correcting errors, inconsistencies, and missing values in the dataset.

Examples:
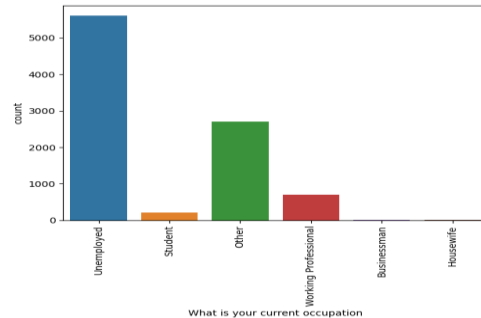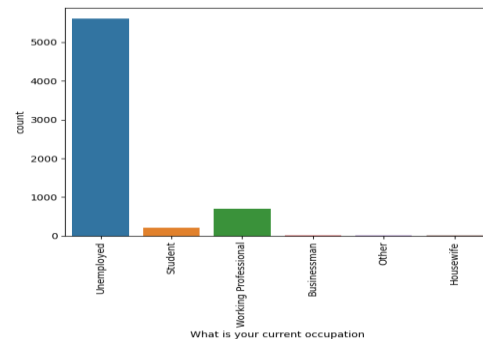
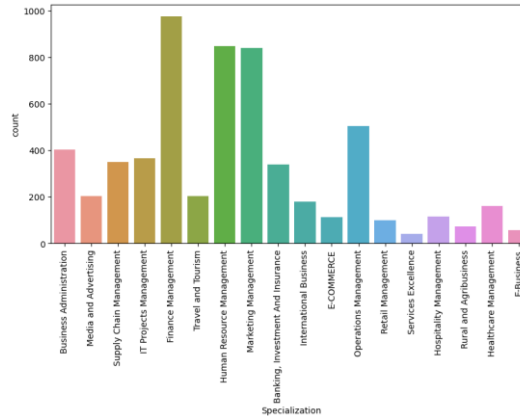Imputing null values of the occupation as Other (Green).

Checking the columns with more than 40% missing values:

- The columns 'Asymmetrique Activity Index,' 'Asymmetrique Profile Index,' 'Asymmetrique Activity Score,' and 'Asymmetrique Profile Score' exhibit approximately 45% null values. Given that these columns are assigned by the sales team after calls and hold no analytical value, they can be safely dropped.
- Both 'How did you hear about X Education' and 'Lead Profile' contain over 70% missing values, making it advisable to remove them from the analysis.
- Similarly, 'Lead Quality' has around 51% missing values and can be dropped.

All the columns with more than 40% null values are of no importance. Hence, we drop them.

After dropping 7 columns, the dataset has 30 columns.

The 'Specialization' column contains 37% missing values. It's plausible that leads may leave this column blank if they are students, lack a specialization, or if their specialization is not among the provided options. To address this, we can introduce a new category, 'Others.'

Cleaning each column of NaN values 30-40% missing values:

1. **'Lead Source'**
   a. Replace google with Google
2. **'Total Visits', 'Page Views Per Visit' and 'Last Activity'**
   a. Replacing with median and mode
3. **'What is your current occupation'**
   a. Imputing with 'Others' category
4. **'What matters most to you in choosing a course'**
   a. Replace with mode
5. **Country , City, Tags**

'Prospect ID' and 'Lead Number' both exhibit no duplicate values, signifying that these columns serve as unique identifiers for each piece of data. As such, their presence does not contribute significantly to our model, and we can safely drop them.

EXPLORATORY DATA ANALYSIS (EDA)

A critical step in understanding the data before building a model, we uncovered the patterns, relationships, and anomalies in the dataset, which is essential for making informed decisions during the model building process. EDA involves summarizing the main characteristics of the data, often with visual methods. Python libraries like Seaborn which is built on top of Matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

According to the problem statement, our target variable is 'Converted.' This variable signifies whether a lead has been successfully converted or not, with the following encoding:

- 0: Not converted into a lead.
- 1: Lead has been successfully converted.

UNIVARIATE ANALYSIS

Univariate Analysis
Univariate Analysis of each column Converted:

 # Converted is the target variable, Indicates whether a lead has been successfully converted (1) or not (0).

Converted =
(sum(leads_df['Converted'])/len(leads_df['Converted'].index))
*100

Converted: 38.53896103896104
The lead conversion rate is about 38%.

Following the univariate analysis, (26 columns) of the Dataset, several columns are identified as not contributing meaningful information to the method. Therefore, it is advisable to eliminate these columns.

'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','A free copy of Mastering The Interview','Country','Do Not Call', and 'Do Not Email'
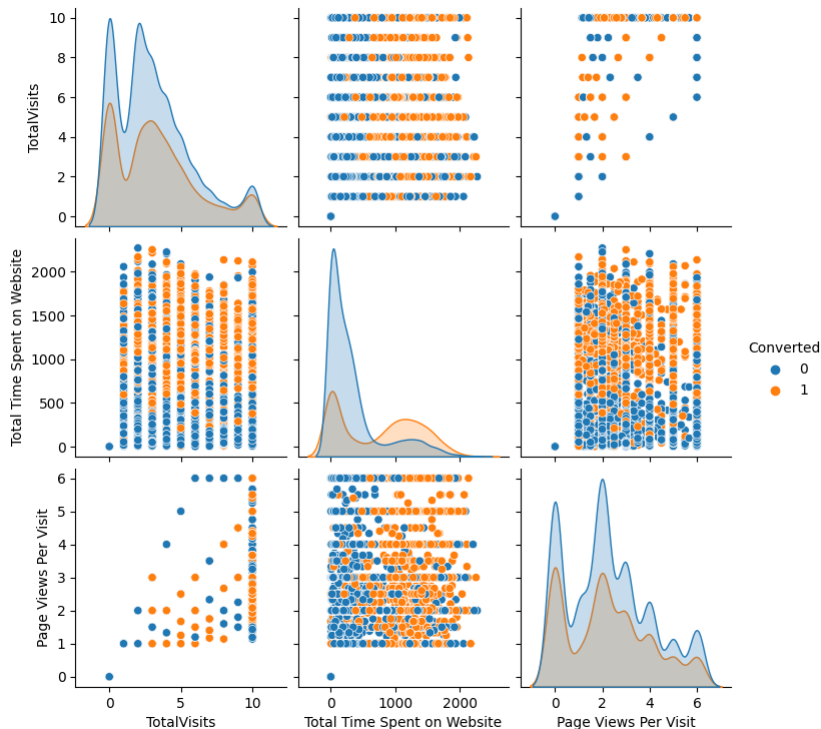
 # Checking the shape of the dataset
leads_df.shape

(9240, 12)
There are 12 columns available after univariate analysis.

BIVARIATE ANALYSIS


```python
# Checking the relation between 'TotalVisits,' 'Page
Views Per Visit,' and 'Total Time Spent on Website',
and leads 'Converted'

leads_num = leads_df[['TotalVisits','Total Time Spent
on Website','Page Views Per Visit','Converted']]

sns.pairplot(leads_num,diag_kind='kde',hue='Converted')

plt.show()
```
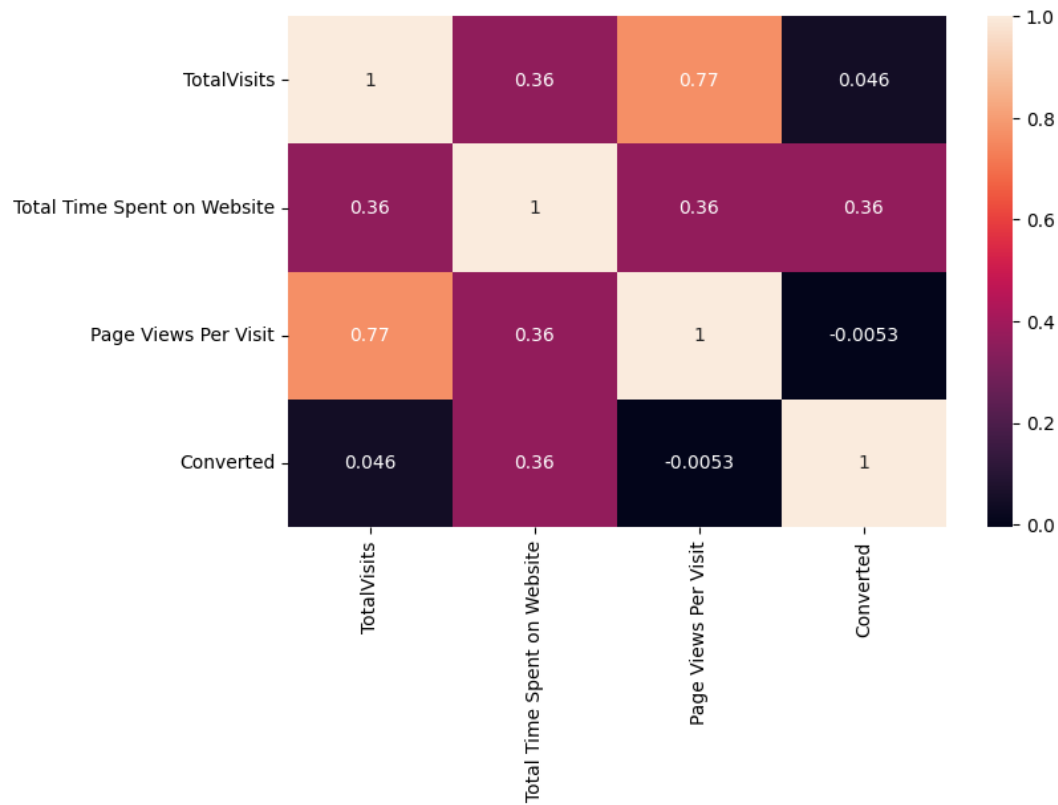
The pairplot shows the relation between 'TotalVisits,' 'Page Views Per Visit,' and 'Total Time Spent on Website', and leads 'Converted'. Where '1' is 'converted' and '0' is 'not converted'.

As the number of 'TotalVisits,' 'Page Views Per Visit,' and 'Total Time Spent on Website' increase, pertentage of positive leads also increases.

'TotalVisits' and 'Page Views per Visit' exhibit a high correlation of 0.77. 'Total Time Spent on Website' demonstrates a correlation of 0.36 with the target variable 'Converted'.

DATA PREPARATION

Step 1 : Creation of dummy variables:

The dataset has 81 columns after the creation of dummy variables and dropping columns for which dummy variables were created.

Step 2 : Feature Scaling

With the domain knowledge, we know the most relevant features that can influence lead conversion are:
1. Total Visits
2. Total Time Spent on Website
3. Page Views Per Visit

```python
# Scaling the features for standardizing features
scaler = StandardScaler()
X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']] = scaler.fit_transform(X_train[['TotalVisits','Total Time Spent on Website','Page Views Per Visit']])
X_train.head()
```

Step 3: Feature Scaling using RFE

```python
#from sklearn.linear_model import LogisticRegression

logreg = LogisticRegression()


#from sklearn.feature_selection import RFE

rfe = RFE(estimator=logreg, n_features_to_select=20)
# running RFE with 20 variables as output


rfe = rfe.fit(X_train, y_train)
```

Python libraries like Scikit-Learn is used for feature scaling.

MODEL BUILDING

```
# Creating and running the model
X_train_sm9 =
sm.add_constant(X_train[col8])
logm9 =
sm.GLM(y_train,X_train_sm9,
family = sm.families.Binomial())
result = logm9.fit()
result.summary()
```

Model 1: Given the high p-value of '**Tags_wrong number given**', it is advisable to remove   this   column.

Model 2: Due to the high p-value of '**Tags_Not doing further education**', it is recommended to eliminate this column.

Model 3: Considering the significantly high p-value of '**Lead Source_Facebook,**' it is advisable to exclude this column.

Model 4: Considering the significantly high p-value of '**Tags_invalid number**', it is better to drop this column.

Model 5: Due to the notably high p-value of '**Last Notable Activity_Had a Phone Conversation**,' it is preferable to exclude this column.

Model 6: Due to the notably high p-value of '**Last Notable Activity_Email Link Clicked**', it is preferable to exclude this column.

Model 7: No p-values are high or need elimination. Dropping column '**Tags_switched off**' to reduce the variables.

Model 8: No p-values are high or need elimination. Dropping column '**Last Notable Activity_Olark Chat Conversation**' to reduce the variables.

In the case of X Education, the company needed a model to assign a lead score to each lead, ultimately aiming to increase the lead conversion rate beyond 80%. After creating 9 models, the **9th model was selected for this purpose.**

This model, a logistic regression, demonstrated an **accuracy of approximately 92.9%** in predicting lead conversions and generated lead scores ranging from 1 to 100 based on the probability of conversion.

*Model 9: As the p-values for all variables are 0, and the VIF values are low for each variable, Model-9 stands as our final model, encompassing a total of **12 variables.***

MODEL EVALUATION

The confusion matrix is used to calculate various performance metrics, such as accuracy, precision, recall, and F1-score, to evaluate the performance of a machine learning algorithm.

```python
from sklearn import metrics

# Confusion matrix
confusion =
metrics.confusion_matrix(y_train_pred_final.Convert
ed, y_train_pred_final.predicted )
print(confusion)
```

[[3805 197]
 [263 2203]]

```python
# Let's check the overall accuracy.

print('Accuracy
:',metrics.accuracy_score(y_train_pred_final.Converted,
y_train_pred_final.predicted))
```
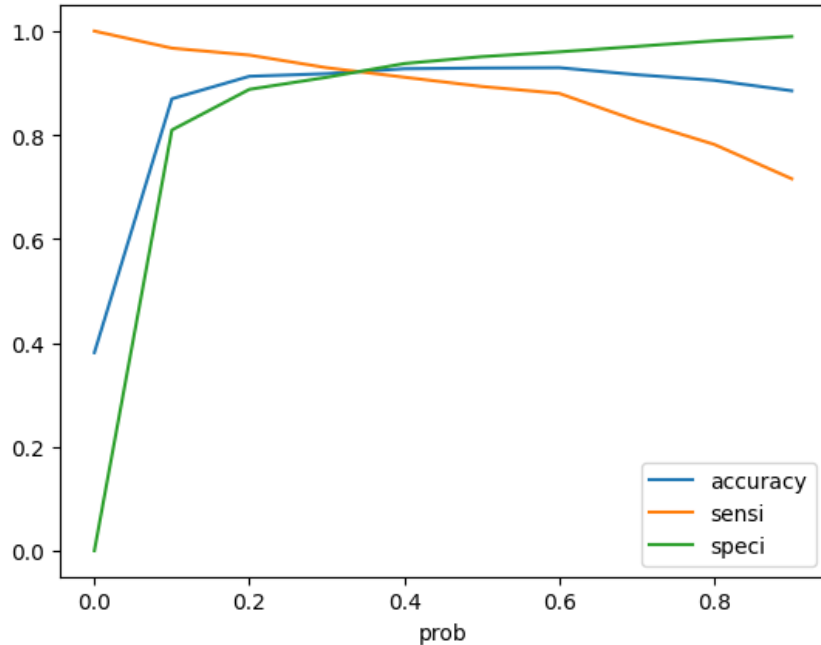
Accuracy : 0.929

Actual/Predicted	Not_converted	Converted
Not Converted	3805	197
Converted	263	2203

```
NOTE:
TP = confusion[1,1] # true positive
TN = confusion[0,0] # true negatives
FP = confusion[0,1] # false positives
FN = confusion[1,0] # false negatives
```

MODEL EVALUATION

Accuracy, Sensitivity and Specificity on Train dataset
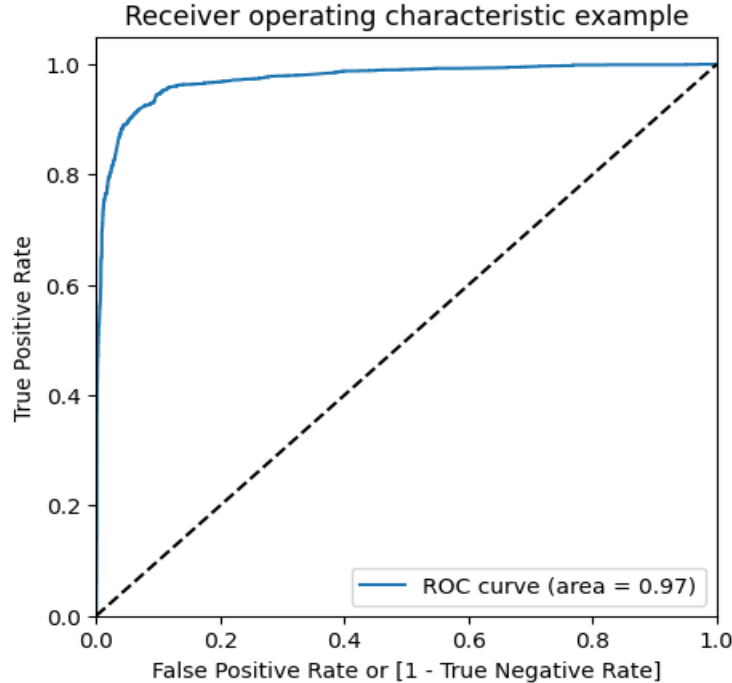

Based on the curve presented, the optimal cutoff probability is determined to be 0.38.

After running the model on the Train Dataset, these are the figures we obtain:

- **Accuracy** : 92.89%
- **Sensitivity** : 89.33%
- **Specificity** : 95.07%

ROC CURVE

Receiver operating characteristic example

ROC curve (area = 0.97)

The Receiver Operating Characteristic (ROC) curve is a graphical representation used to evaluate the performance of a binary classification model. It illustrates the trade-off between Sensitivity (the ability to correctly identify positive instances) and 1 - Specificity (the rate of false alarms) across different classification thresholds. The Area Under the ROC Curve (AUC-ROC) provides a single metric to quantify the model's overall discriminatory power, with a higher AUC-ROC indicating better performance.

With a higher area under the ROC curve (0.97), our model is considered good. The graph shows the performance of a classification model at all classification thresholds.

The accuracy of the model can be further assessed using metrics, such as specificity, sensitivity, precision and others, related to the problem of lead scoring and conversion rate.

Sensitivity: The sensitivity of the logistic regression model **0.893** means that the model is **good at identifying true positive leads (TP)**. A higher sensitivity indicates that the model is better at identifying potential leads that eventually convert into paying customers.

Specificity: The specificity of the logistic regression model **0.951** means that the model is **good at identifying true negative leads (TN)**. A higher specificity indicates that the model is better at filtering out leads that are unlikely to convert into paying customers.

False Positive Rate: The false positive rate is **0.049**, which means that the model occasionally predicts a converted lead when it is not actually converted (FP). A lower false positive rate indicates that the model is **better at avoiding unnecessary follow-ups with potential leads** that are not likely to convert.

Positive Predictive Value: The positive predictive value is **0.918**, which means that the model is **good at predicting converted leads (TP)**. A higher positive predictive value indicates that the model is better at identifying potential leads that are likely to convert into paying customers.

Negative Predictive Value: The negative predictive value is **0.935**, which means that the model is good at predicting non-converted leads (TN). A higher negative predictive value indicates that the model is better at filtering out leads that are unlikely to convert into paying customers.

PRECISION AND RECALL

Precision, also termed Positive Predictive Value, denotes the proportion of relevant results among the predicted positive outcomes is **0.918 or 91.8%**

Recall, also referred to as Sensitivity, represents the percentage of total relevant results accurately classified by the algorithm is **0.893 or 89.3%**
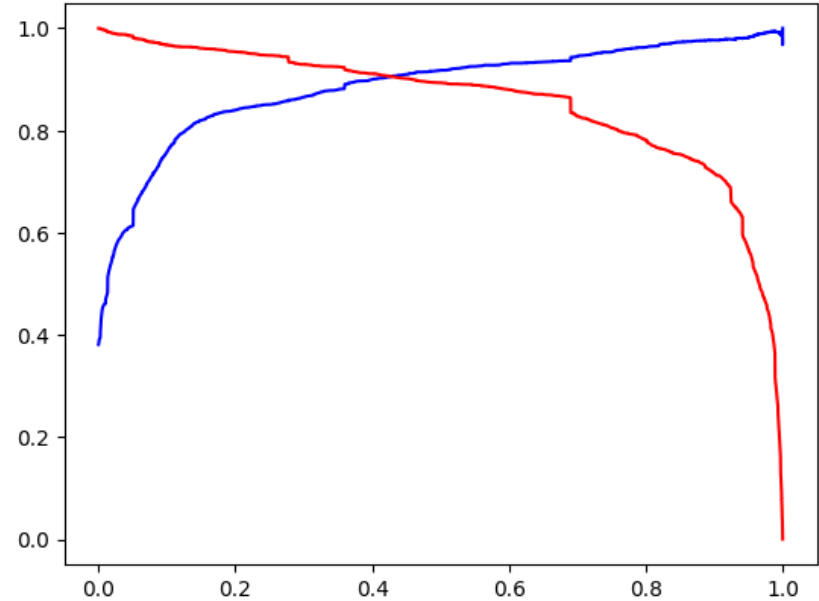
MODEL PREDICTING

`y_pred_final.head()`

Prospect ID	Converted	Converted_prob	final_predicted
4269	1	0.678683	1
2376	1	0.988495	1
7766	1	0.894766	1
9199	0	0.002554	0
4359	1	0.922763	1

Observations:

After running the model on the Train Dataset, these are the figures we obtain:

- Accuracy : 92.89%
- Sensitivity : 89.33%
- Specificity : 95.07%

After running the model on the Test Dataset, these are the figures we obtain:

- Accuracy : 91.85%
- Sensitivity : 96.07%
- Specificity : 89.08%

CONCLUSION

- A lead scoring model estimates how close a lead is to making a purchase by assigning points when they show interest in your business. By using a lead scoring model, X Education can find the potential customers who are the best fit for their product, focus on contacts who are most likely to convert, and know where a prospect is in the customer journey.

- The benefits of a lead scoring model include increased revenue cycles, increased return on investment (ROI), and optimized marketing and sales alignment.

- In summary, a lead scoring model is a powerful tool for finding potential customers, making sales, and saving time doing it all.

RECOMMENDATIONS

1. Actively engaging customers or leads who fill out forms signifies potential opportunities for conversion.

2. Targeting working professionals is advisable due to their higher conversion probability and potentially better financial capabilities for service payments compared to those without specified occupations.

3. Leads with 'Last Activity' marked as 'SMS Sent' exhibit a heightened conversion rate, emphasizing the need to prioritize them in targeted marketing efforts.

4. Analyzing the behavior of customers spending more time on the website can significantly improve user experience and boost conversion rates. Therefore, the company should emphasize creating compelling content and ensuring user-friendly navigation to encourage prolonged website engagement.

5. Understanding the popularity of various specializations allows for tailored course offerings and marketing campaigns. Providing targeted content and resources, especially for popular specializations such as Management, can attract and retain customers within those specific fields. These insights contribute to a more effective and targeted approach in customer engagement and overall business strategies.

THANK YOU