
LEAD-SCORING-CASE-STUDY-
UPGRAD
By Ayush Saxena

AGENDA

READ AND UNDERSTAND THE DATA

DATA CLEANING AND PREPARATION

PREPARE THE DATA FOR MODELLING

- MODEL BUILDING
- DUMMY VARIABLE CREATION
- TEST-TRAIN SPLIT
- SCALING
- LOOKING AT THE CORRELATIONS

MODEL EVALUATION

- FINDING THE OPTIMAL CUTOFF
- MAKING PREDICTIONS ON THE TEST SET

PRECISION-RECALL VIEW

- PRECISION AND RECALL TRADEOFF
- MAKING PREDICTIONS ON THE TEST SET

STEPS USED

The steps are broadly:

1. Read and understand the data
2. Clean the data
3. Prepare the data for Model Building
4. Model Building
5. Model Evaluation
6. Making Predictions on the Test Set


UNDERSTANDING THE DATA

ABOUT THE DATA SET & CLEANING DATA SET

The dataset consists of 9240 leads with 37 columns. It includes a mix of categorical, numeric, and binary variables. Many columns have missing values, requiring handling. Categorical variables need encoding, possibly using one-hot encoding. The 'Converted' column serves as the target variable, indicating lead conversion. Preprocessing and exploratory analysis are necessary steps before applying machine learning models.

ABOUT THE DATA SET & CLEANING DATA SET

Observations

1. Data Overview:
   - Dataset contains 9240 leads with 37 columns.

2. Data Types:
   - Includes 4 float64, 3 int64, and 30 object data types.

3. Missing Values:
   - Many columns have missing data, notably Lead Source, TotalVisits, Page Views Per Visit, and others.

4. Categorical Variables:
   - Several columns represent categorical variables such as Lead Origin, Lead Source, Last Activity, and more.

5. Numeric Variables:
   - Numeric variables include TotalVisits, Total Time Spent on Website, and Page Views Per Visit.

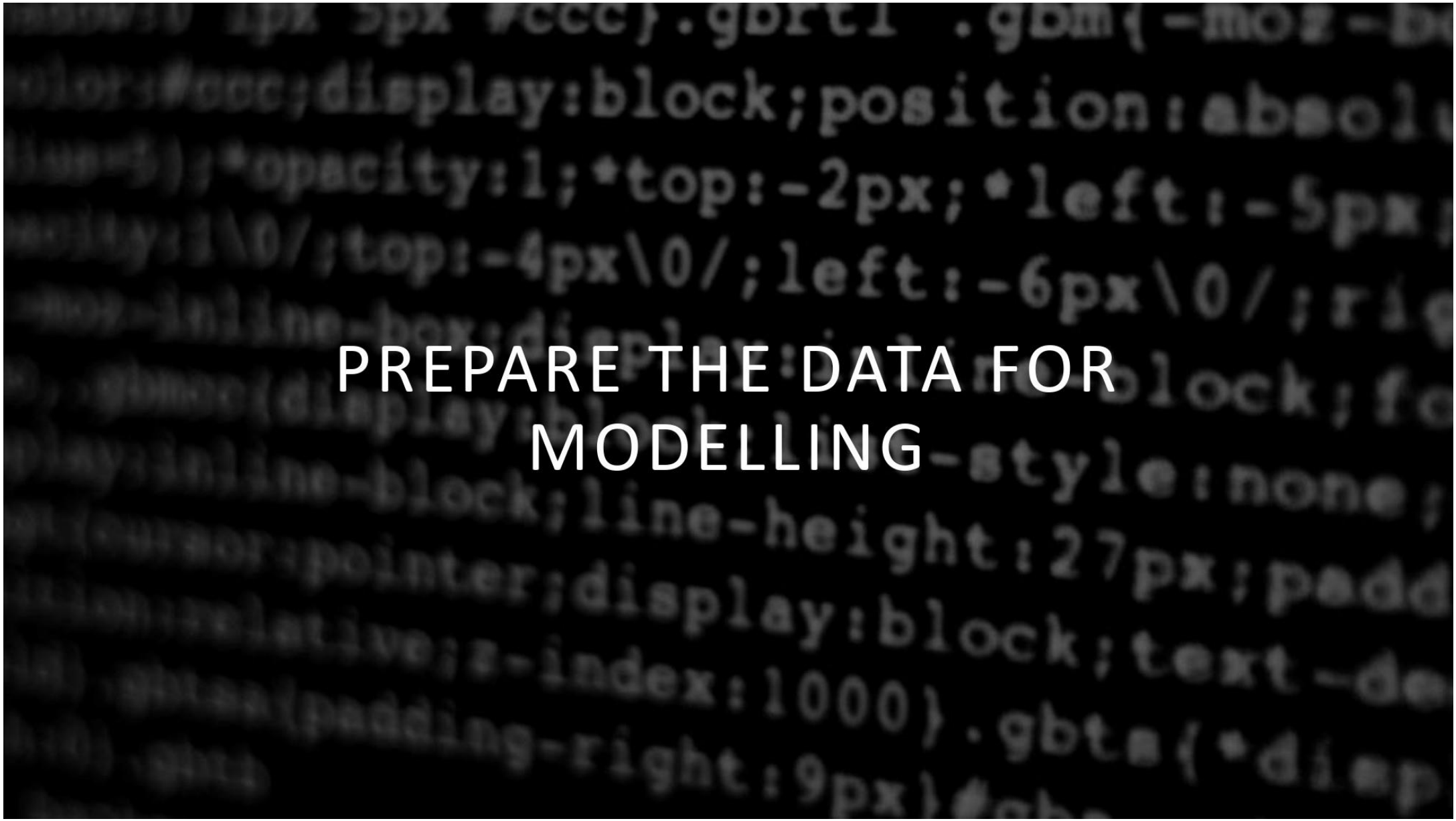
ABOUT THE DATA SET & CLEANING DATA SET

Observations

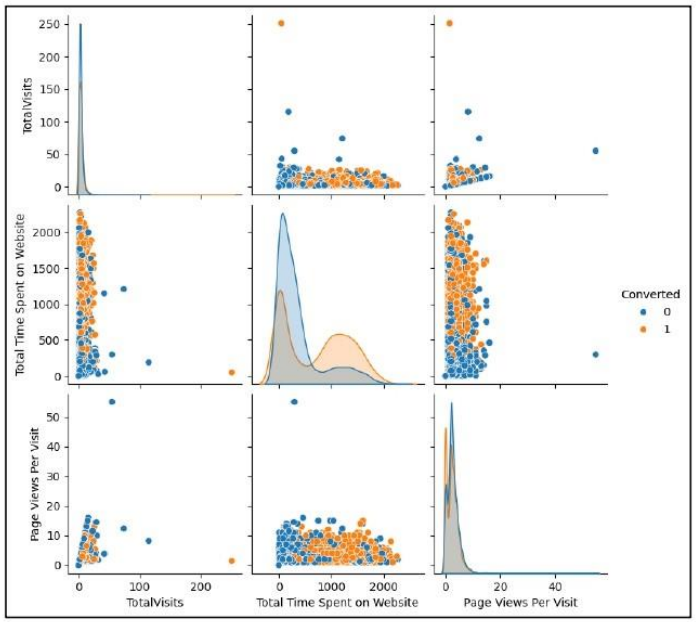
6. Binary Variables:
   - Binary variables like Do Not Email, Do Not Call, Search, etc., are present.

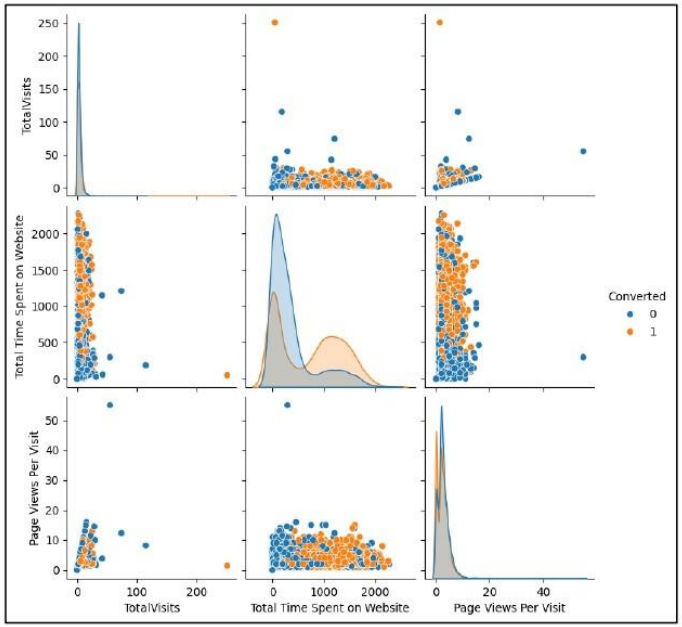
7. Target Variable:
   - 'Converted' column indicates whether a lead was converted or not.

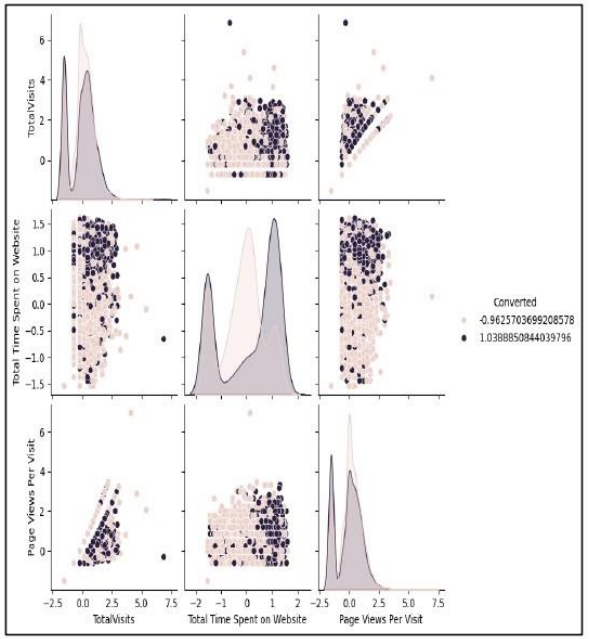
8. Next Steps:
   - Handle missing values.
   - Encode categorical variables.
   - Preprocess data for machine learning.
   - Conduct exploratory data analysis for deeper insights.


PREPARE THE DATA FOR
MODELLING

DATA MODELING







Interpretation:

•This pairplot helps in visually exploring the relationships between different variables and their impact on lead conversion.

•It can help identify which features might have a strong influence on conversion and how they interact with each other.

DATA MODELING

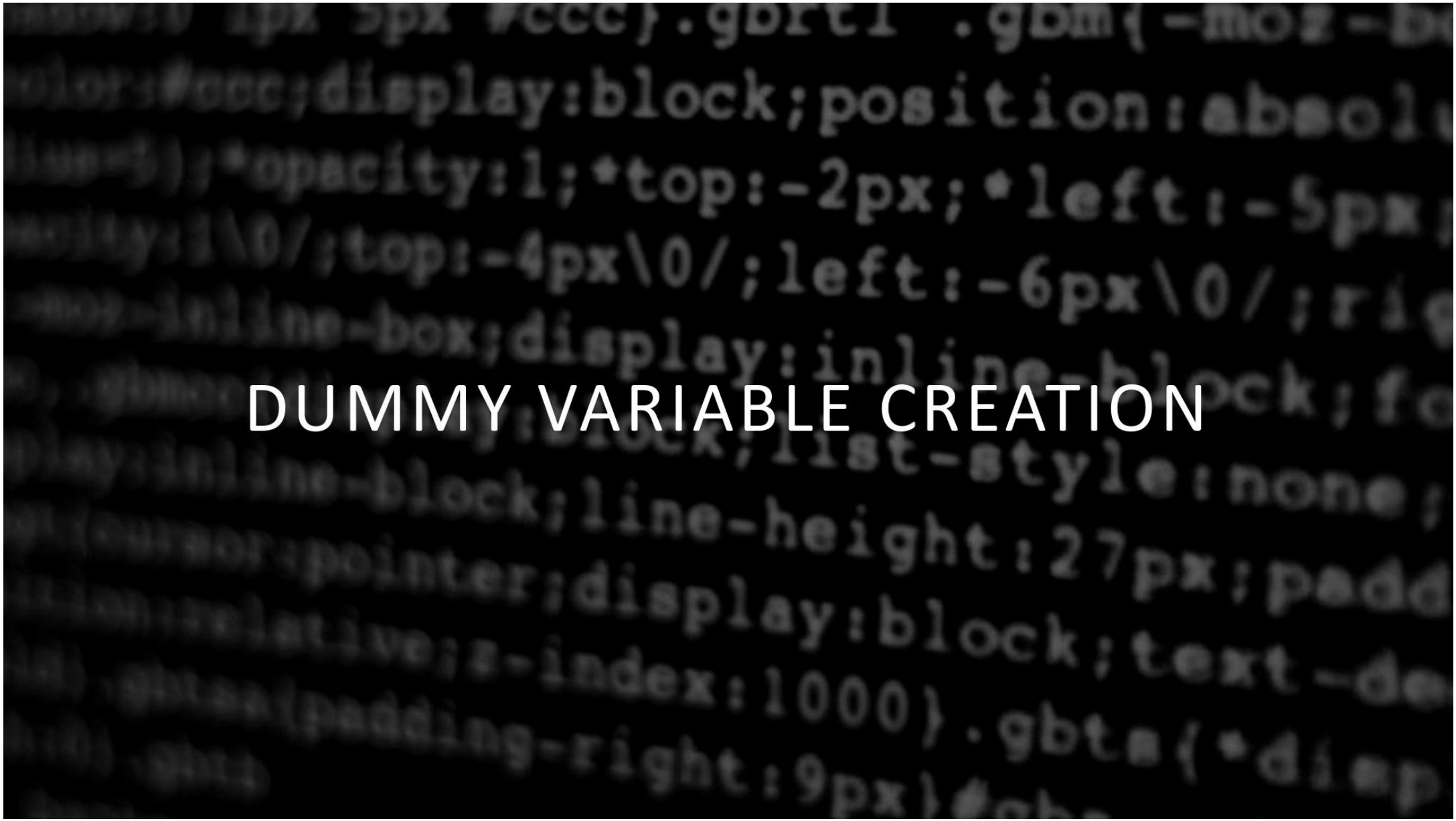
1. Pairplot Analysis with Original Data:
   - Pairplot analysis is performed on the entire dataset (`xleads`) with 'Converted' as the hue.
   - It provides a visual comparison of different features against each other, especially with respect to conversion status.
   - Kernel density estimation (KDE) is used for diagonal plots.
   - The plot aims to understand the distribution of variables and their relationship with conversion status.

2. Pairplot Analysis with Selected Features:
   - Another pairplot analysis is performed, this time focusing on a subset of features (`TotalVisits`, `Total Time Spent on Website`, `Page Views Per Visit`, `Converted`).
   - Similar to the previous plot, KDE is used for diagonal plots and 'Converted' is used as the hue.
   - This plot specifically examines the relationship between selected features and conversion status.

3. Power Transformation:
   - Power transformation using the `PowerTransformer` from `sklearn.preprocessing` is applied to the selected subset of features (`xedu`).
   - Power transformation is often used to stabilize variance and make the data more Gaussian-like.
   - The transformed data is stored in `transformedxedu`

4. Pairplot Analysis with Transformed Data:
   - Pairplot analysis is performed on the transformed dataset (`transformedxedu`) with 'Converted' as the hue.
   - KDE is used for diagonal plots.
   - This plot helps visualize the effect of power transformation on the distribution of features and their relationship with conversion status.

In summary, the snippets perform exploratory data analysis (EDA) by visualizing the distribution and relationship of features with respect to conversion status, both with the original data and after applying power transformation to selected features. These visualizations aid in understanding the data distribution and identifying potential patterns or correlations with the target variable.


DUMMY VARIABLE CREATION

DATA MODELING

Interpretation:

•identifying the categorical variables within the dataset.

1. Identifying Categorical Variables:

- It first identifies columns in the dataset with data type 'object', indicating categorical variables.

2. Creating Dummy Variables:

- Dummy variables are created using `pd.get_dummies()` for specific categorical columns like 'Lead Origin', 'Lead Source', etc.
- The parameter `drop_first=True` is used to avoid multicollinearity issues by dropping the first level of each categorical variable.

3. Creating Dummy Variable for 'Specialization':

- Another set of dummy variables is created separately for the 'Specialization' column.
- The level 'Select' is dropped explicitly, likely because it's considered redundant or uninformative.

4. Concatenating Dummy Variables:

- The newly created dummy variables are concatenated with the original dataframe `xleads`.

5. Dropping Original Categorical Columns:

- The original categorical columns for which dummy variables were created are dropped from the dataframe to avoid redundancy.
- These columns include 'Lead Origin', 'Lead Source', 'Do Not Email', 'Last Activity', 'Specialization', 'What is your current occupation', 'A free copy of Mastering The Interview', and 'Last Notable Activity'.

6. Final Dataset View:

- Lastly, it displays the first few rows of the modified dataframe `xleads` to review the changes made.


SPLITTING THE DATA

SPLITING THE DATA

Interpretation:

The next step is to split the dataset into training an testing sets.

1. Importing Library:
- **It imports the `train_test_split` function from the `sklearn.model_selection` module.**

2. Defining Features and Target:
- **All feature variables except 'Converted' are stored in the variable `X`.**
- **The target variable 'Converted' is stored in the variable `y`.**

3. Splitting the Dataset:
- **The dataset is split into training and testing sets using `train_test_split()` function.**
- **`X_train` and `y_train` contain 70% of the data for training.**
- **`X_test` and `y_test` contain the remaining 30% of the data for testing.**
- **The parameter `random_state=100` ensures reproducibility of the split.**

4. Final Dataset Split:
- **The dataset is now ready for training and testing machine learning models with 70% of the data allocated for training and 30% for testing.**

This code snippet essentially prepares the dataset for machine learning tasks by splitting it into features (X) and the target variable (y), and then further splitting them into training and testing sets.


SCALING THE DATA

SCALING THE DATA

Interpretation:

Now there are a few numeric variables present in the dataset which have different scales.

1. Importing MinMax Scaler:
- **It imports the `MinMaxScaler` class from the `sklearn.preprocessing` module.**

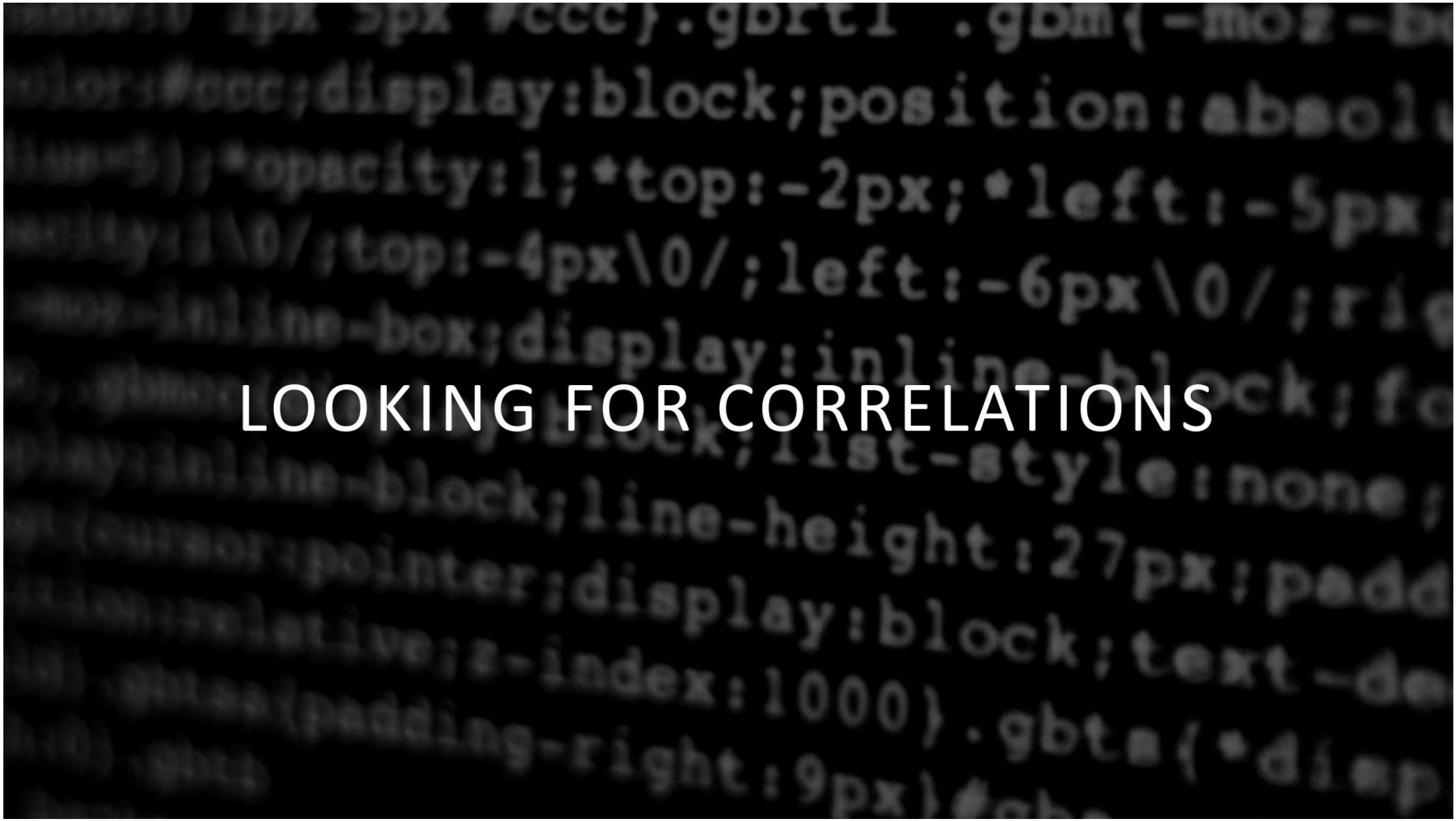
2. Scaling Numeric Features:
- **The MinMaxScaler is applied to scale the three numeric features present in the dataset: 'TotalVisits', 'Page Views Per Visit', and 'Total Time Spent on Website'.**
- **The scaler is initialized as `scaler`.**

3. Scaling Transformation:
- **`fit_transform()` method is used to scale the numeric features in the training set (`X_train`).**
- **The MinMaxScaler scales the features to a range between 0 and 1.**

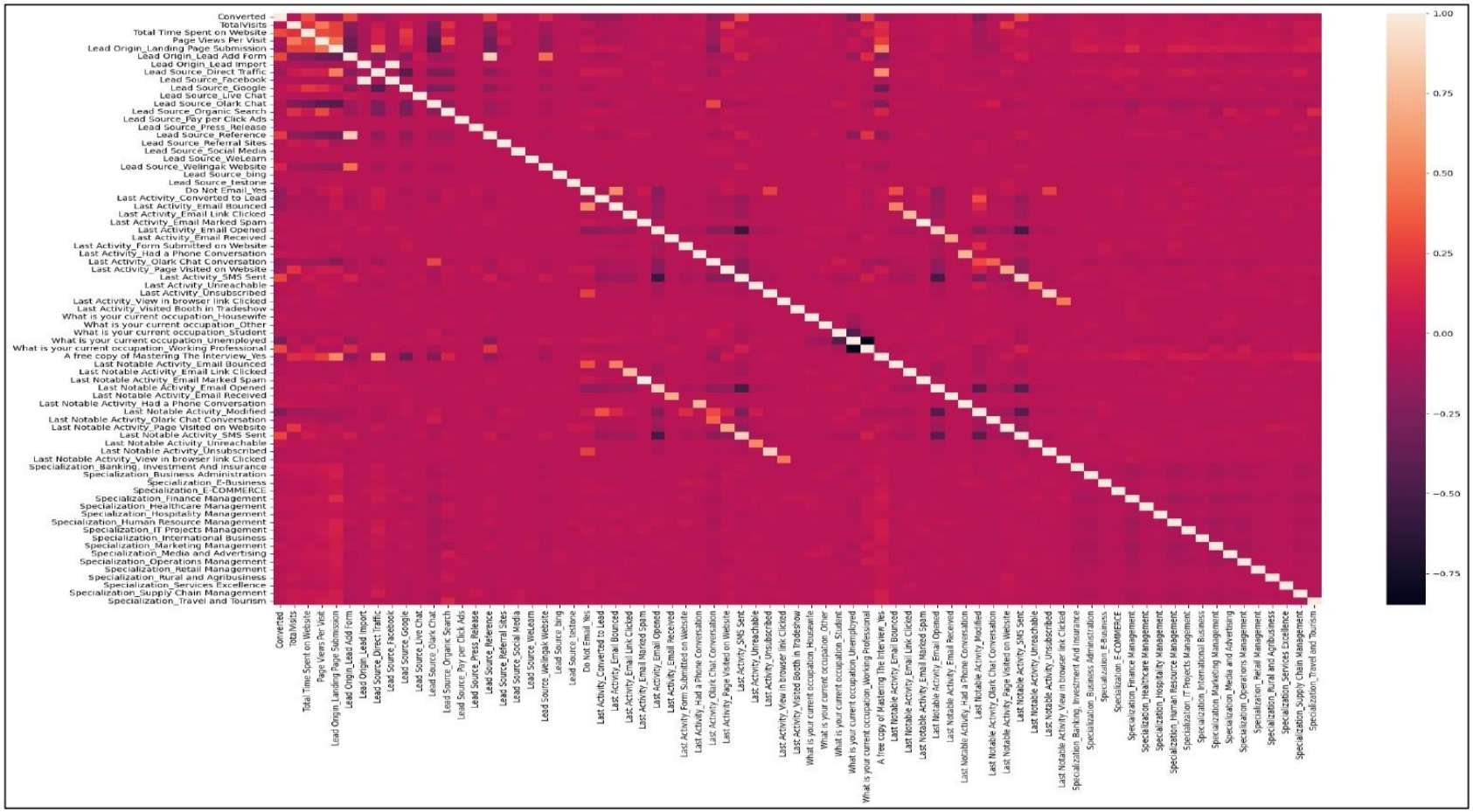
4. Update Training Data:
- **The scaled features are replaced in the original training dataset (`X_train`) to reflect the scaled values.**

5. Displaying Updated Training Data:
- **The first few rows of the updated training dataset are displayed using `head()` to observe the changes after scaling.**

In summary, this performs feature scaling on the numeric features in the training dataset using MinMaxScaler, ensuring that all features are on the same scale.


LOOKING FOR CORRELATIONS

CORRELATIONS



CORRELATIONS

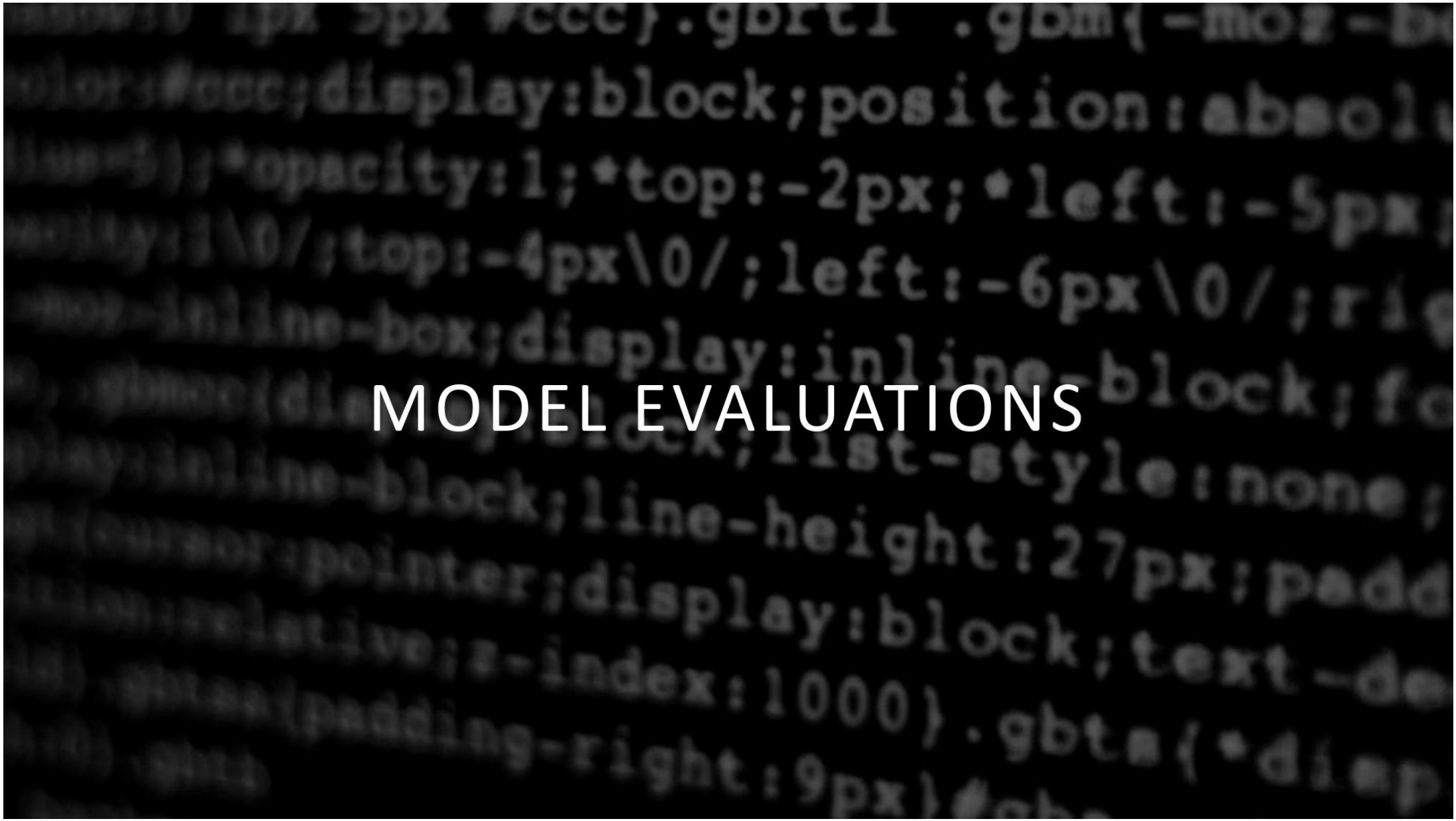
Interpretation: There is no correlation between the variables

1. Heatmap Visualization:
- **The code generates a heatmap using Seaborn's `heatmap` function.**

2. Correlation Matrix:
- **The heatmap represents the correlation matrix of the entire dataset `xleads`.**
- **Correlation coefficients between variables are visualized as colors in the heatmap.**

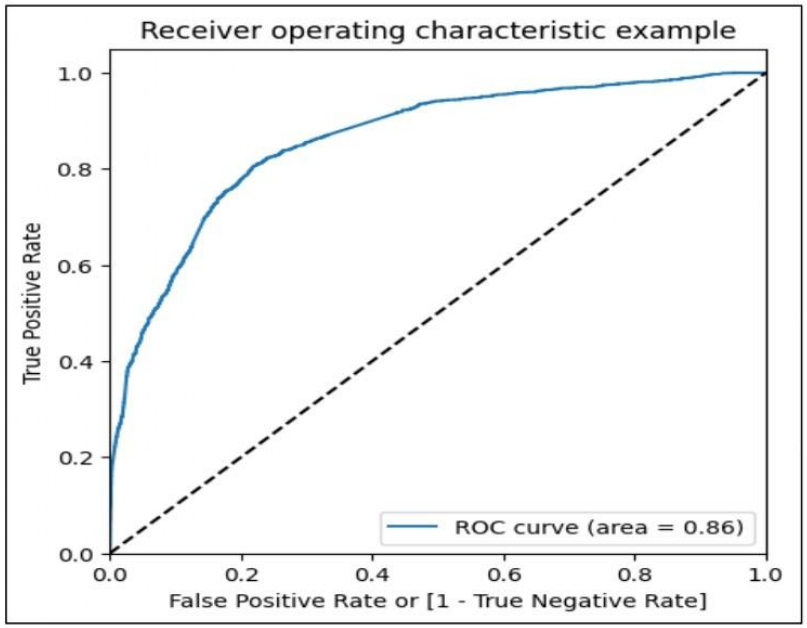
3. Interpretation:
- **Darker shades indicate stronger correlations, while lighter shades indicate weaker correlations.**
- **It helps identify patterns of correlation between different variables, aiding in feature selection and understanding relationships within the dataset.**

4. Figure Size:
- **The size of the heatmap figure is set to be 25 inches wide and 15 inches tall for clarity and readability.**

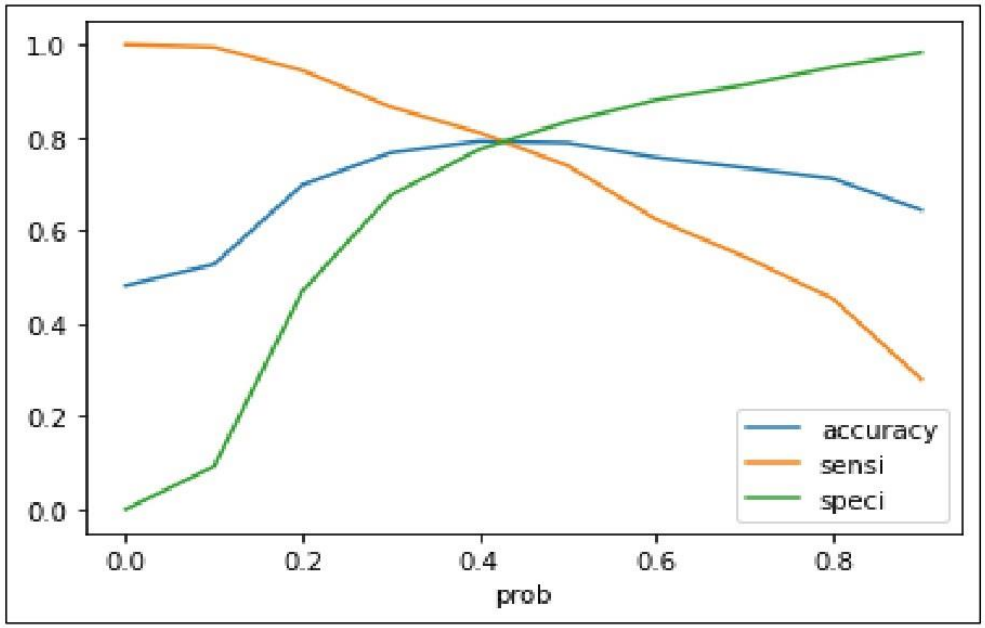

MODEL EVALUATIONS

ROC CURVE

Interpretation: 0.42 is the tradeoff between Precision and Recall - Thus we can safely choose to consider any Prospect Lead with Conversion Probability higher than 42 % to be a hot Lead



The area under the curve of the ROC is 0.86 which is quite good. So we seem to have a good model. Let's also check the sensitivity and specificity tradeoff to find the optimal cutoff point.



As you can see that around 0.42, you get the optimal values of the three metrics. So let's choose 0.42 as our cutoff now.


MAKING PREDICTIONS ON THE
TEST SET

OBSERVATIONS

Train Data:
Accuracy : 80%
Sensitivity : 77%
Specificity : 80%

Test Data:
Accuracy : 80%
Sensitivity : 77%
Specificity : 80%

Final Features list:

- Lead Source Olark Chat
- Specialization Others
- Lead Origin_Lead Add Form
- Lead Source_Welingak Website
- Total Time Spent on Website
- Lead Origin Landing Page Submission
- What is your current occupation Working Professionals
- Do Not Email

SUMMARY

1. Identifying Potential Prospects:

- Prioritize leads based on factors such as 'TotalVisits', 'Total Time Spent on Website', and 'Page Views Per Visit' that significantly contribute to the probability of lead conversion.
- Sorting out the best prospects helps in focusing efforts on leads with higher conversion potential.

2. Lead Nurturing Strategy:

- Maintain a comprehensive list of leads to keep them informed about new courses, services, job offers, and future opportunities.
- Tailor communication and information based on the interests and preferences of individual leads.
- Implement a personalized approach to nurture leads effectively, increasing the chances of conversion.

3. Focus on Converted Leads:

- Concentrate efforts on converted leads by organizing question-answer sessions to gather relevant information.
- Engage in further inquiries and appointments to understand the intentions and mindset of leads regarding online courses.
- Utilize insights gained from interactions to tailor offerings and communication, enhancing lead engagement and conversion rates.

The strategy involves prioritizing potential prospects, nurturing leads through personalized communication, and focusing efforts on converted leads by gathering insights and tailoring offerings to increase the likelihood of lead conversion.

THANK YOU