Below is the approach I followed to create the Jupyter Notebook:

Data Analysis:

1. First, the given pickle data is read with a pandas dataframe. The dataset has many missing values that will not add any importance to the model understanding. So the columns containing more than 50% of missing values are dropped. So only one column *example1* is remaining.
2. Then the duplicate rows are dropped.
3. The column has nested dictionary values, so the model can not be trained directly on such dataset. So the JSON normalized function is used from pandas.
4. After normalizing the dataframe, it still consists of the JSON type of data. So again it is normalized similarly.
5. Once the dataframe without JOSN is generated, there are columns with repeating names, so names are duplicated for the columns.
6. To solve this issue, I have split the dataframe with columns into three different dataframes, from where it is being repeated. After that, all three dataframes are merged into a single dataframe.
7. The newly created dataframe also has missing values in entire rows, so I have removed it using *dropna()* method. Also, one column `meeting_link_clicked` contains only 4 rows, so that column is also dropped.

Data Visualization:

The column *opened* is plotted with *True* and *False* values.

Text Preprocessing:

As the data to train is textual data, the preprocessing is done using NLTK library. One functionis created which does the cleaning of data as below:

1. The lower casing of the characters
2. Removal of extra spaces
3. Removal of punctuation marks
4. Steeming and stop word removal

The function to split text into words is done.

Model Training:
How did I decide on the target variable?

When the mail is received by the receiver, I am trying to analyze that in which type of content, the user is opening the link and visiting the site.

Once the data is prepared, it is time to train the model.

1. First, the data is split into train and test keeping the test dataset size at 33%.
2. The feature extraction from the text corpus is done with the TF-IDF method for training and testing data.
3. To train the model with text data, the *Multinomial Naive Bayes* algorithm is used as it works well when dealing with textual data.

Model Prediction:

To predict and test the model, an accuracy score metric is used on test data.