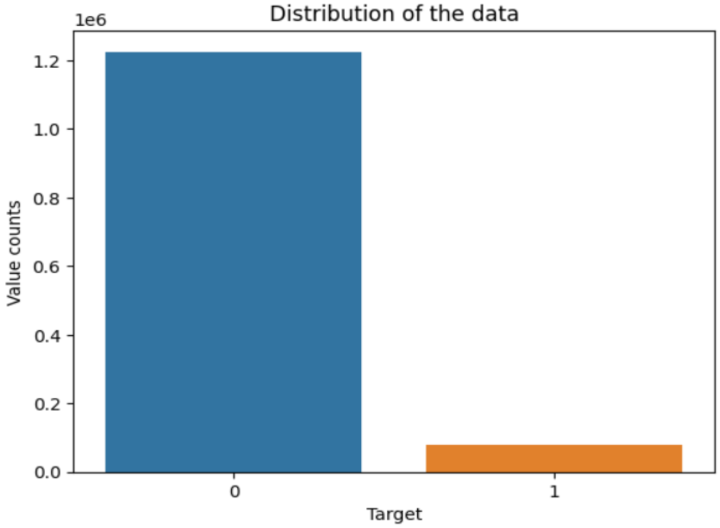
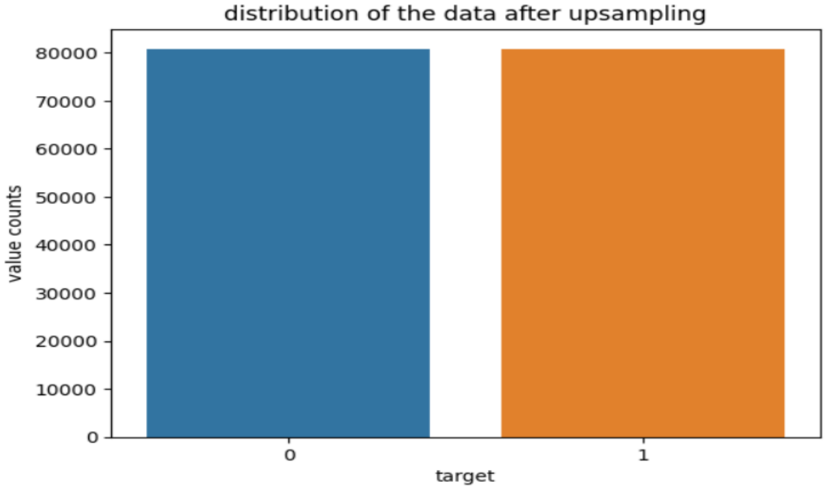
Problem Statement: Detects whether a question asked in Quora is sincere or not.

Dataset: The Dataset is collected from Kaggle. The dataset includes questions that were asked and labeled as sincere or not. It contains over 13 Lakh labeled data points in the training and test set consisting of around 3 Lakh unlabeled examples. Each data point in the training set has a unique id and the question text along with a binary label where '0' represents sincere and '1' represents 'insincere'.

DATE IMBALANCE: The Dataset is highly imbalanced where 93% of the data belongs to target 0 and only around 7% to target 1 as shown below:



To deal with this imbalance, I use Random undersampling. It involves randomly selecting examples from the majority class and deleting them from the training dataset. The resulting dataset will have an equal number of data points for both the classes.



TEXT Preprocessing: Data preprocessing is a crucial step in machine learning, especially for NLP tasks. It involves several stages:

1. Tokenization: Breaking down text into individual words or tokens. This step converts a sentence into a list of words, making it easier to apply NLP techniques.

2. Lowercasing: Standardizing all text to lowercase ensures that the algorithm treats words like "Great," "great," and "gReat" identically, eliminating case-sensitive variability.
3. Removing Stop Words and Punctuation: Stop words (e.g., "and," "is," "do") are frequently used in language but don't add significant meaning to the analysis. Punctuation marks are also often removed as they generally don't contribute to understanding the semantic meaning of text in many NLP tasks.
4. Stemming: This process reduces words to their root form. For instance, "learn," "learning," and "learnt" are stemmed to a common base word. Stemming helps to decrease the size of the vocabulary and treats different forms of the same word as one, simplifying further analysis.

I have used NLTK library which is a standard python library with prebuilt functions for natural language processing.

FEATURE EXTRACTION: Feature Engineering is a very key part of Natural Language Processing. as we all know algorithms and machines can’t understand characters or words or sentences hence we need to encode these words into some specific form of numerical in order to interact with algorithms or machines. we can’t feed the text data containing words /sentences/characters to a machine learning model. I have used the following methods of feature extraction with text data:

1. TF-IDF: TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. Refer the link for detailed explanation : https://medium.com/@cmukesh8688/tf-idf-vectorizer-scikit-learn-dbc0244a911a

DIMENSIONALITY REDUCTION :In the project, dimensionality reduction is performed using Truncated SVD, as PCA is not suitable for sparse matrices like those obtained from TF-IDF. Due to memory constraints, a subset of 3,000 data points is sampled from the original dataset for this process. This approach allows for effective dimensionality reduction while managing resource limitations.

MACHINE LEARNING MODELS USED: I experimented with the following machine learning algorithms:

1. Logistic regression
2. Logistic regression with L1 regularization
3. SVM Classifier
4. Decision Tree
5. Random Forest