**Data Analytics For Information System Final Project :**

**Disaster Tweet Identification**

# Introduction

To predict if the tweet is about a real disaster or not.

* The provided dataset has a target label which indicates 1 for the real disaster tweet and 0 for the false report of disaster tweet.
* After EDA and Feature Engineering, we pass it through various algorithms for the final prediction of the disaster tweets on the dataset.

# Dataset description

* The tweets dataset has 5 labels named id, keyword, location, text, target.
* The id label acts like a primary key which acts like an identifier for each tweet.
* Text label includes the content of the tweet.
* Location is the source of the tweet.
* Keywords are the words from the text which contains the tweet but most of the keywords are blank or incorrectly marked.
* Target has 0s and 1s which indicate whether the tweet is real or fake.

# Exploratory data analysis

The graphical description of information and data is known as data visualization. Data visualization tools make it easy to examine and comprehend trends, outliers, and patterns in data by engaging visually like charts, graphs, and maps.

# 1. Bar plot

We have plotted the bar graph for the number of tweets falling under the real and fake category. It is clear that the data is highly unbalanced. There are more fake tweets than the real ones.

# 2. Histogram

The length of the tweet has a limit of 280 characters. So we have created a new feature length which consists of the length of the corresponding tweet. Then we have plotted the histogram of the number of tweets against the new feature length falling in two categories: real and fake. We can see that the length of a tweet between 120-140 characters has a high chance of being real.

# Data preprocessing

1. **Data cleaning**: Data can contain a lot of useless and missing information. Data cleaning is carried out in order to handle this component. It entails dealing with missing data, noisy data, and so on.

* Handling Null Values: In anydataset, there are always few null values. It doesn’t really matter whether it is a regression, classification or any other kind of problem, no model can handle these NULL or NaN values on its own so we need to come about.
* Standardization: It is another intrinsic preprocessing step. we transform our values such that the mean of the values is replaced by NULL or NAN.
* Handling Categorical Variables: It is another intrinsic aspect of Machine Learning.Categorical variables are basically the variables that are discrete and not continuous.
* One-Hot Encoding: So in One-Hot Encoding what we essentially do is that we create ’n’ columns where n is the number of unique values that the nominal variable can take.
* Multicollinearity: It occurs in our dataset because we have features that are firmly dependent on each other.
* Missing Data: This circumstance occurs when some data in the data is missing. It can be dealt with in a variety of ways.

The text label can contain the attributes as URLs, Tags, Emoticons, HTML Tags,

Stopwords and Useless characters.

The URLs, Tags and HTML tags were handled by creating a function which replaces these entities with a space.

Incase of emoticons, these were already handled in the csv file as they got converted into characters.

The stopwords generally are unimportant ones as the,is, and etc, which are most common words in a sentence. These are high frequency words with little semantic weight and so will not be of much use while training the model. Such words were removed from the text label.

Stemming is done in order to reduce the words to its root form. This reduces the size of the dictionary. It is nothing but removing the last few characters from a word.

The above cleaning was done on the keywords and location as well.

Word-Cloud - Word-cloud shows all the words in the text label in different sizes, meaning, bigger the size, more frequent is the word and smaller the size, less number of times the word has occurred in a tweet. We merged the columns keywords, location and text in order to create a word-cloud. We have imported Word-cloud because we wanted a visual representation of most of the words in the tweet.

2. **Data Transformation**: This step is conducted to transform the data into a format that can be used in the mining process. We are using a function called MinMaxScaler and what it does is that it retains the shape of the original distribution of data. It doesn't change the information embedded in the original data. It doesn't reduce the importance of outliers.

The default range for the feature MinMaxScaler is 0 to 1.

# 3. Data Reduction

In data mining we deal with large amounts of data. While dealing with large amounts of data, analysis becomes more heavy. So we use data reduction techniques to get rid of this hurdle. Its goal is to simply to improve storage efficiency by lowering data storage and analysis expenses. The above changes were made in the training dataset through the help of user defined functions which are mentioned in the EDA of the code. After having made the changes in the training dataset, similar parameterswere passed in the test dataset for homogeneity.

**Models**

# 1. Naive Bayes

The Bayes theorem provides the basis for a collection of supervised machine learning classification algorithms known as Naive Bayes. It's a simple categorization technique with a lot of power. They're useful when the inputs' dimensionality is high. The Naive

Bayes Classifier can also be used to solve complex classification issues.

The Bayes Theorem is used to create Naive Bayes Classifiers. The high independence assumptions between the features are one of the assumptions made. These classifiers make the assumption that the value of one feature is unrelated to the value of any other characteristic. Naive Bayes Classifiers are particularly efficient in supervised learning situations. To estimate the parameters needed for classification, naive Bayes classifiers require a small amount of training data. Naive Bayes Classifiers are easy to develop and execute, and they can be used in a variety of real-world settings.

# 2. Gaussian Naive bayes

Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution.It is independent of Y I or Xi I or both I Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution. To build a simple model, assume the data is characterized by a Gaussian distribution with no covariance (independent dimensions) between the parameters. This model can be fitted by simply calculating the mean and standard deviation of the points within each label, which is all that is required to construct a distribution of this type.

# 3. Bernoulli Naive Bayes

Bernoulli Naive Bayes is a member of the Naive Bayes family. It only accepts binary data. The most basic example is determining whether or not a word appears in a document for each value. That is a rudimentary model. Bernoulli may get better results in circumstances where counting the frequency of words is less critical. To put it another way, we must count every value binary term occurrence feature, such as whether a word appears in a document or not. Rather than finding the frequency of a word in the document, these attributes are utilised.

# 4. Complement Naive Bayes

Complement Naive Bayes is a hybrid of the Multinomial Naive Bayes and Complement Naive Bayes algorithms. On unbalanced datasets, Multinomial Naive Bayes does not perform well. Imbalanced datasets are those in which the number of examples of one class exceeds the number of examples of other classes. This indicates that the number of examples is not evenly distributed. This type of dataset can be challenging to work with because a model could easily overfit the data in favor of the class with the most examples.

# 5. Multinomial Naive Bayes

In Natural Language Processing, the Multinomial Naive Bayes method is a common Bayesian learning approach (NLP). Using the Bayes theorem, the program estimates the tag of a text, such as an email or a newspaper piece. It assesses the likelihood of each tag for a given sample and returns the tag with the highest probability. The Naive Bayes classifier is made up of several algorithms, all of which have one thing in common: each feature being classified is unrelated to any other feature. The presence or absence of one trait has no bearing on the inclusion or exclusion of another.

# 6. SVM

Back vectors are information focuses that are closer to the hyperplane and have an impact on the hyperplane's position and introduction. We maximize the classifier's edge by utilizing these back vectors. The hyperplane's position will be changed in the event that the back vectors are erased. These are the focuses that will help us in building our SVM.

# Final Training Results

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Logistic**  **Regression** | **Gaussia**  **n Naive**  **Bayes** | **Bernoulli**  **Naive Bayes** | **Complement**  **Naive Bayes** | **Multinomial**  **Naive Bayes** | **RBF**  **Kernel** | **Linear**  **SVC** | **Random**  **Forest**  **Classifier** |
| **Average**  **Accuracy** | **81.00%** | **75%** | **80%** | **86%** | **82%** | **83%** | **82%** | **80%** |

**Above results were obtained after running the model for 20+ times.**

**Conclusion:** The model which provided us with most average accuracy was Complement Naive Bayes which is 86%.