**Group No.: G4**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **Email @st.niituniversity.in** |
| 1. | AMAN KHANDELWAL | Aman.Khandelwal |
| 2. | SAGNIK MITRA | Sagnik.Mitra |
| 3. | SHASHWAT SHAH | Shashwatm.Shah |

**Reference Paper Title: Multi-source multi-class fake news detection**

**Authors:** Hamid Karimi, Proteek Chandan Roy, Sari Saba-Sadiya, and Jiliang Tang

**Goal of your work, pointing out differences, if any with the paper:**

“Fake news”, this definition focuses on two dimensions: the intentionality (very difficult to prove) and the fact that the story is made up. This implies that honest mistakes are not considered to be fake news. Finding ways to determine fake news from real news is a challenge most Natural Language Processing folks want to solve. There is significant difficulty in doing this properly and without penalizing real news sources.

We are comparing Multinomial Naive Bayes on a bag-of-words features as well as on a Term Frequency-Inverse Document Frequency features. We are also comparing a Passive Aggressive linear classifier using the TF-IDF features. The resulting accuracy should range relatively higher.

**Work done so far including who has done what (you may use more space if required):**

* We explored datasets, used a Pandas DataFrame and check the shape, head and applied any necessary transformations.
* We extracted training data, separated the labels and set up training and test datasets.
* Used bag-of-words and Term Frequency–Inverse Document Frequency (TF-IDF) to extract features. Using longer text will hopefully allow for distinct words and features for the real and fake news data.

Data Exploration: Aman Khandelwal

Data Extraction: Sagnik, Shashwat

**Plan of work for rest of the semester including who will do what:**

* Now that we have training and testing data, we can build the classifiers. To get a good idea if the words and tokens in the articles had a significant impact on whether the news was fake or real, we begin by using CountVectorizer and TfidfVectorizer.
* We will be comparing models using Confusion Matrices since it is easier compare and read with it, so we have to use scikit-learn library to create confusion matrices.
* Finally we will be testing the linear models with TF-IDF vectorizers.