Fake News Detection System

Final Report

***AIDI 2004-02-AI in Enterprise Systems***

***Course Facilitator – Ms. Noopa Jagadeesh***

**Submission by**

Atul Kumar (100800135)

Mohseen Mehmood Shaikh (100799844)

Harsha Patel (100802701)

Nasir JanAngel (100825909)

**Submission Date: August 11, 2021**

Problem Statement:

* Fake News is intentionally written to mislead readers to believe false information, which makes it difficult and nontrivial to detect based on news content​
* Exploiting the information is challenging in and of itself as users’ social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. ​

Objectives:

Ever read a piece of news which just seems bogus? We all encounter such news articles, and instinctively recognise that something doesn’t feel right. Because of so many posts out there, it is nearly impossible to separate the right from the wrong. Here, we are not only talking about spurious claims and the factual points, but rather, the things which look wrong intricately in the language itself.

There are two ways of claiming that some news is fake or not: First, an attack on the factual points. Second, the language. The former can only be done through substantial searches into the internet with automated query systems. It could be an overwhelming task, especially for someone who is just getting started with data science and natural language processing.

The latter is possible through a natural language processing pipeline followed by a machine learning pipeline. It is how we would implement our **fake news detection project**in Python. It is another one of the problems that are recognized as a machine learning problem posed as a natural language processing problem.

Dataset:

There are many datasets out there for this type of application, but we would be using the one mentioned [here](https://www.kaggle.com/c/fake-news/data). The training data contains about 25000+ news feeds with two target labels: Unreliable or reliable and the testing data with around 5800+ records are not labelled. The dataset also consists of the title of the specific news piece.

Dataset Url: <https://www.kaggle.com/c/fake-news/data>

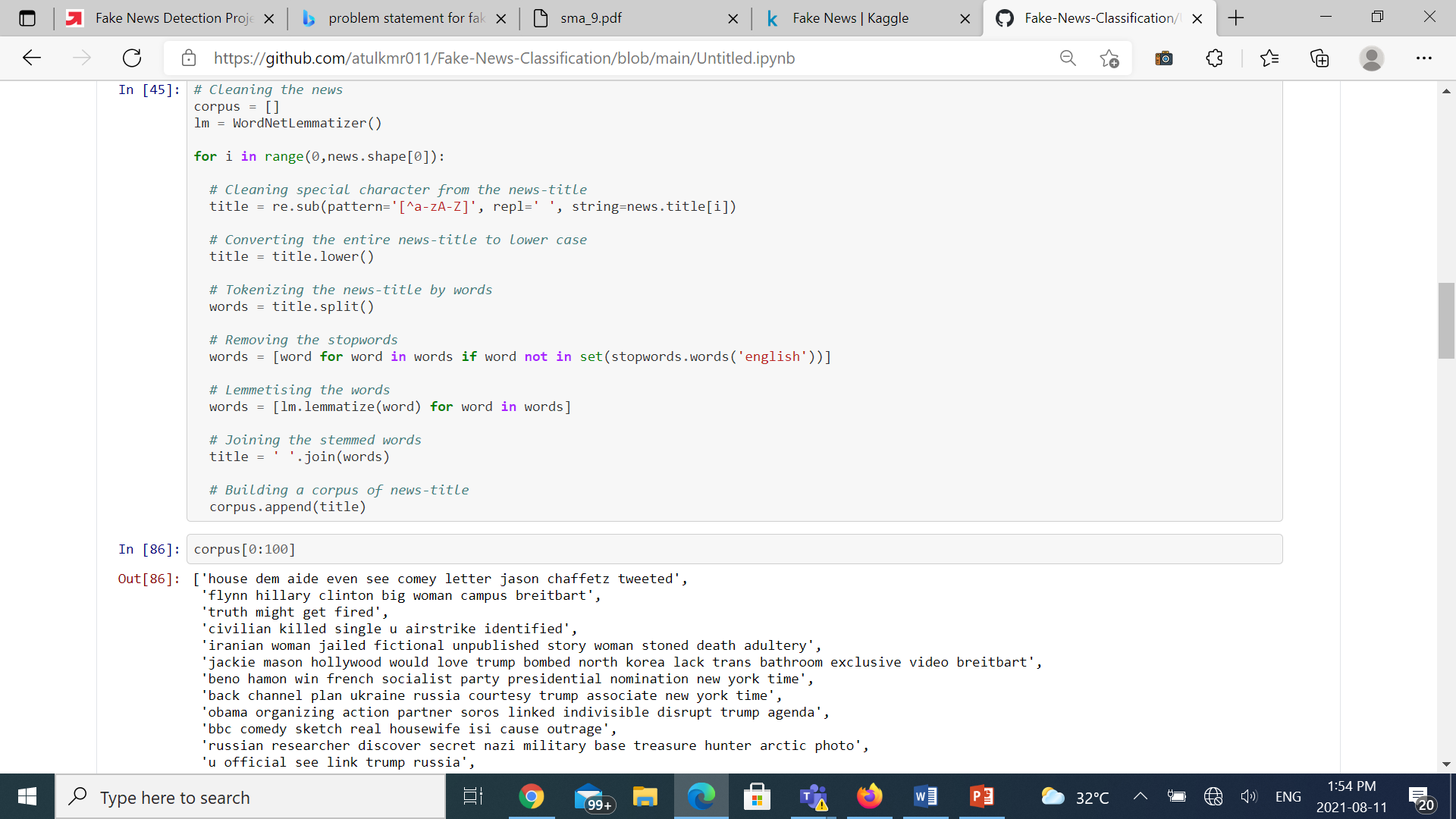
Data Description:

train.csv: A full training dataset with the following attributes:

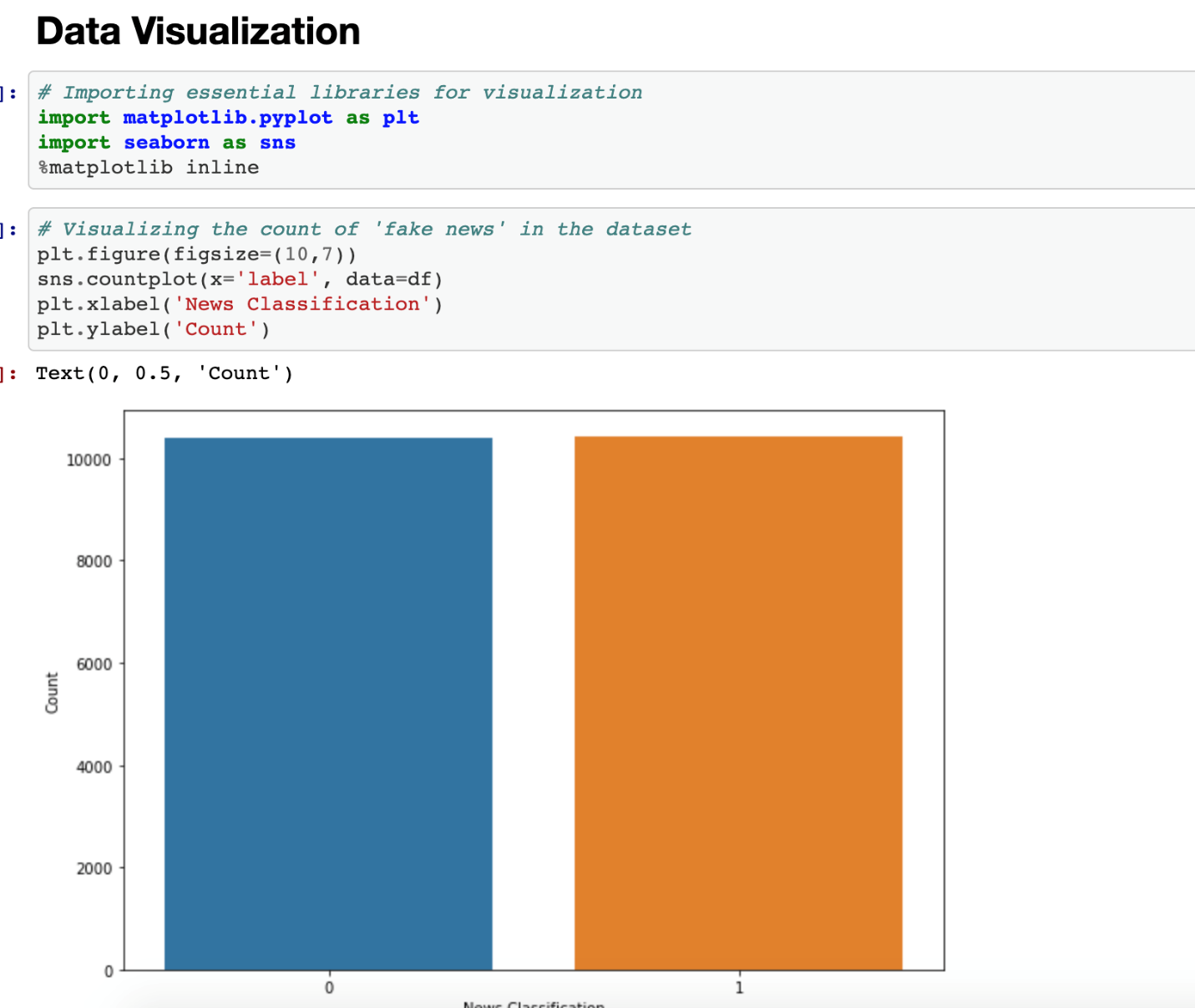
* id: unique id for a news article
* title: the title of a news article
* author: author of the news article
* text: the text of the article; could be incomplete
* label: a label that marks the article as potentially unreliable
* 1: unreliable
* 0: reliable

test.csv: A testing training dataset with all the same attributes at train.csv without the label.

Data pre-processing:

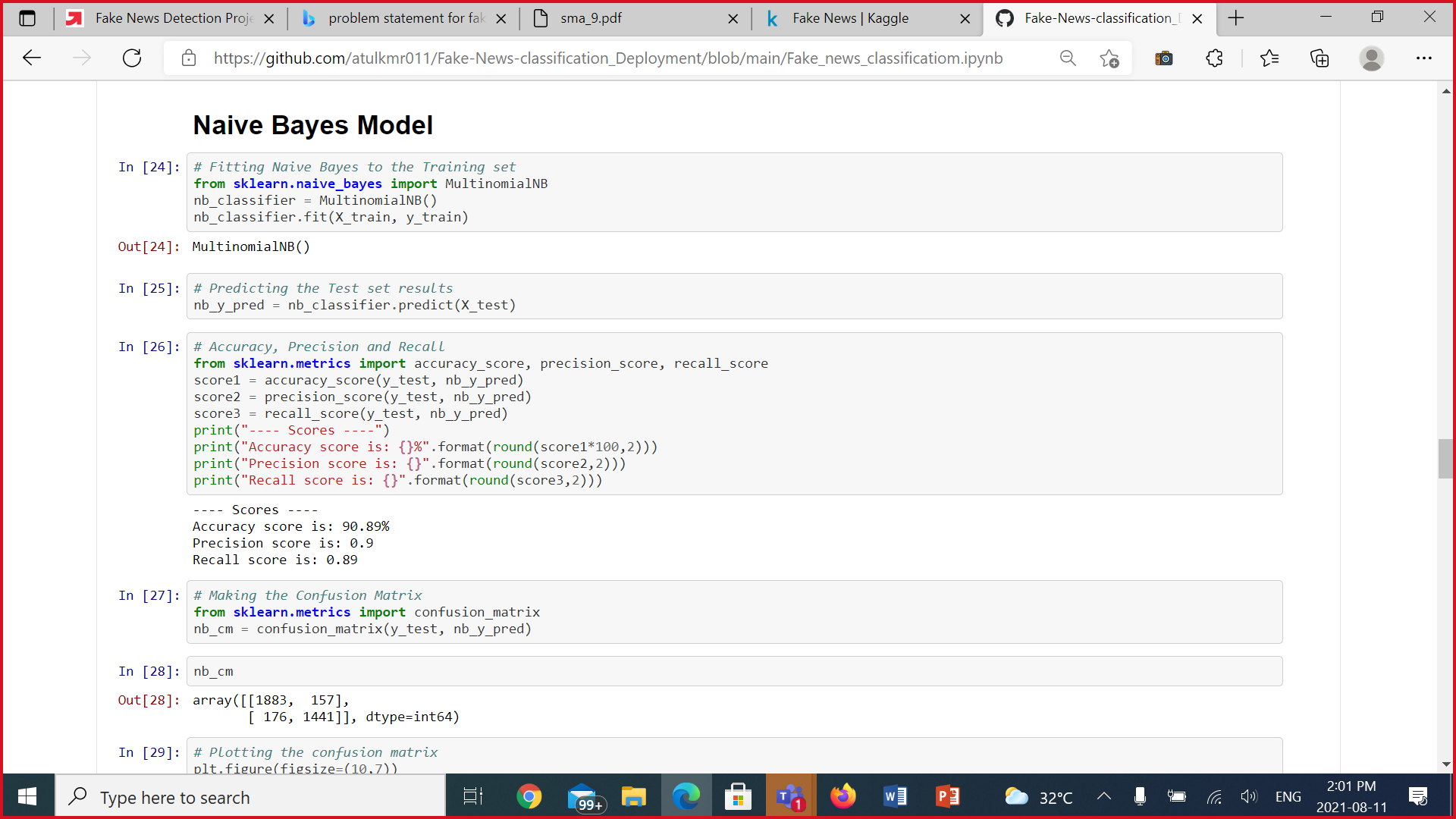


* Cleaning
  + We have used regex and NLTK to clean the values in our dataset.
* Tokenization
  + To tokenize the words, we created a dictionary of words of their frequencies with which they appear in a string line.
  + For e.g., “How are you my friend?” becomes [‘How’,’are’,’you’,’my’,’friend’,’?’]
* Lemmatization
  + We have removed all the stop words and performed lemmatization
* Encoding
  + Representation of categorical variables as binary vectors.

We have 50% real news data and 50% fake news data

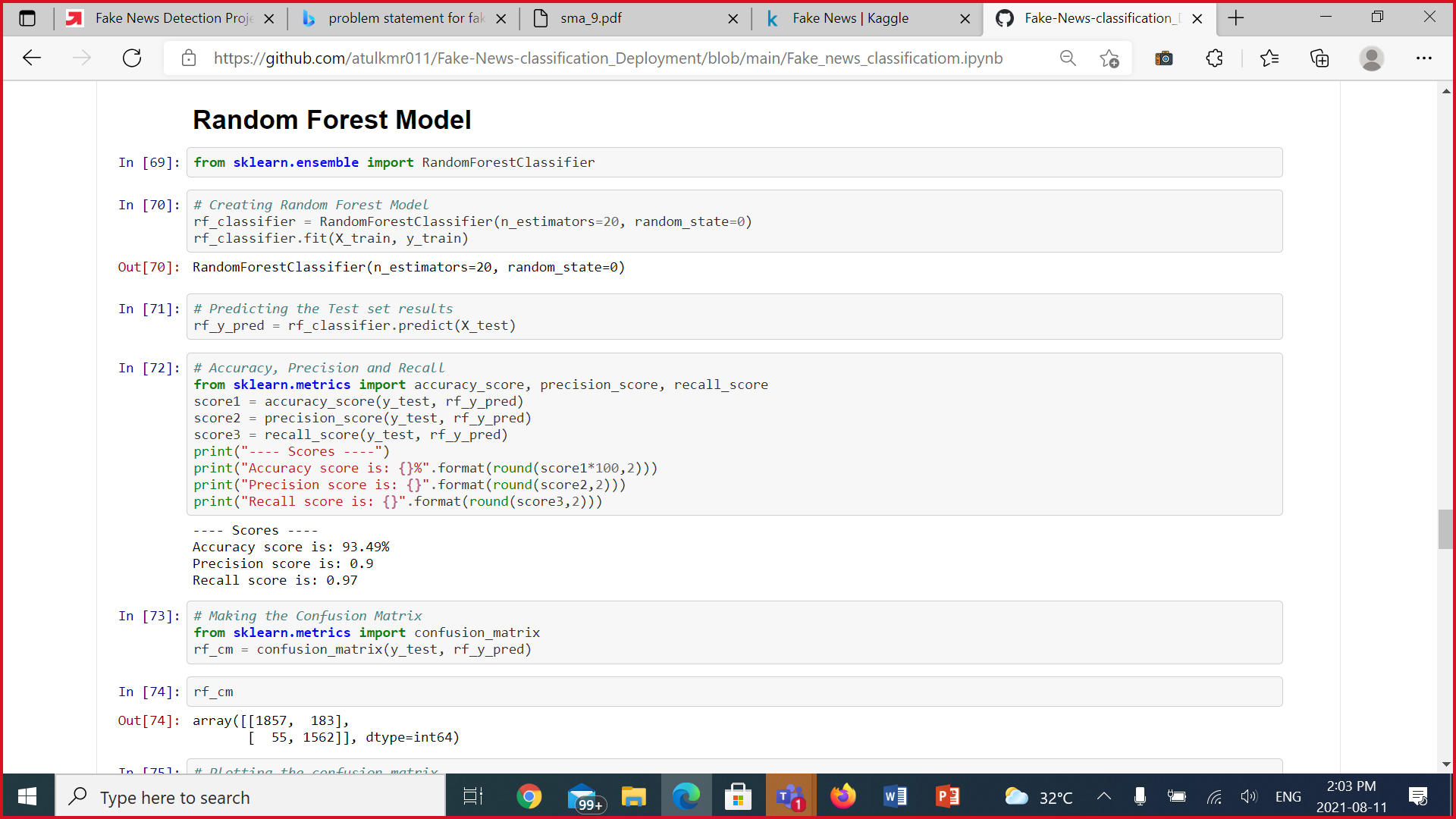
Model Approach:

Model 1: Naïve Bayes Model



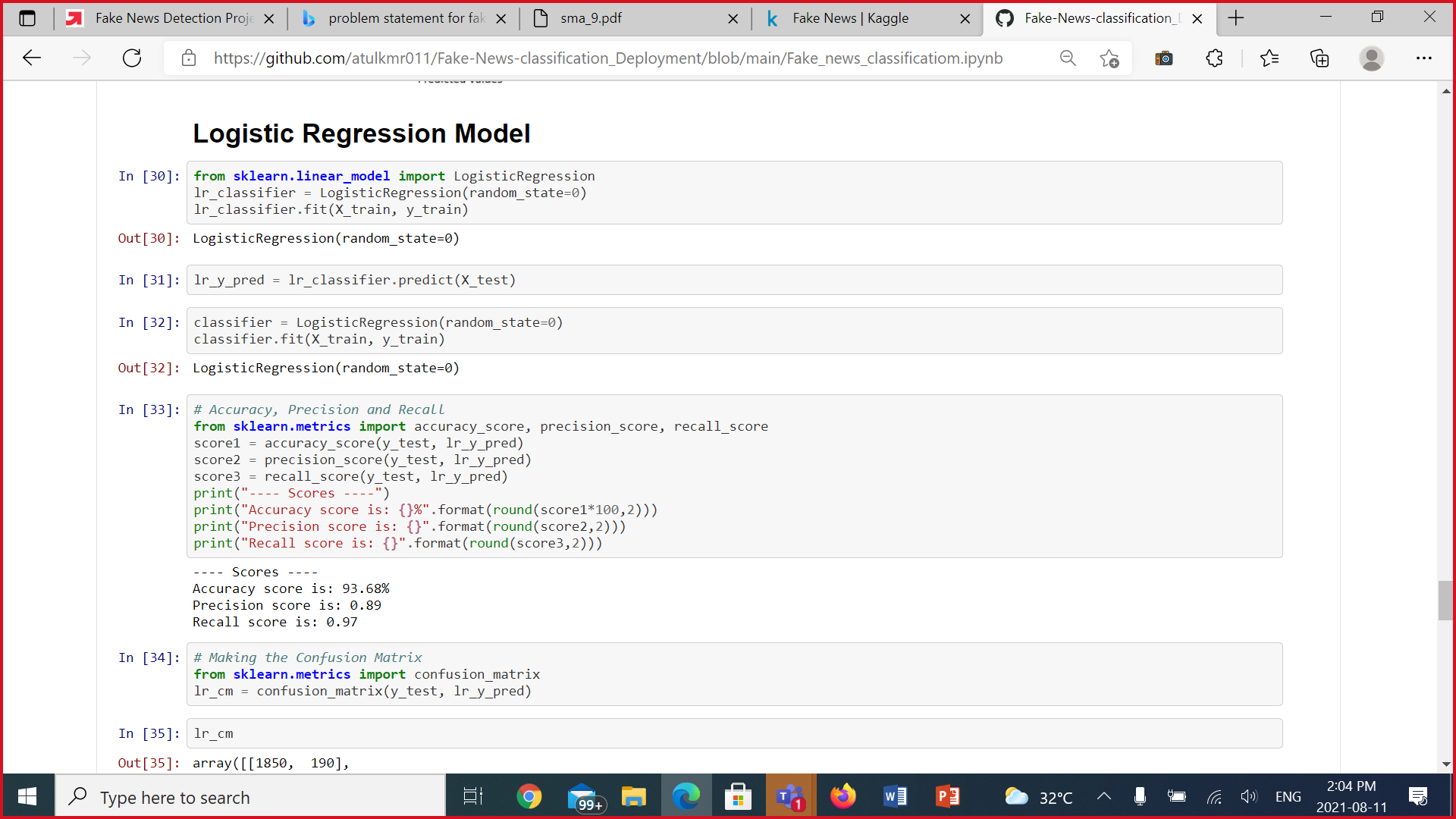
Naïve Bayes Model gave an appreciable accuracy of 90.89%. Then we tried Random forest classification in order to get a higher accuracy score when compared to Naïve Bayes Model.

Model 2: Random Forest Classification



Random Forest Classification model gave a very good accuracy of 93.49%

Model 3: Logistic Regression



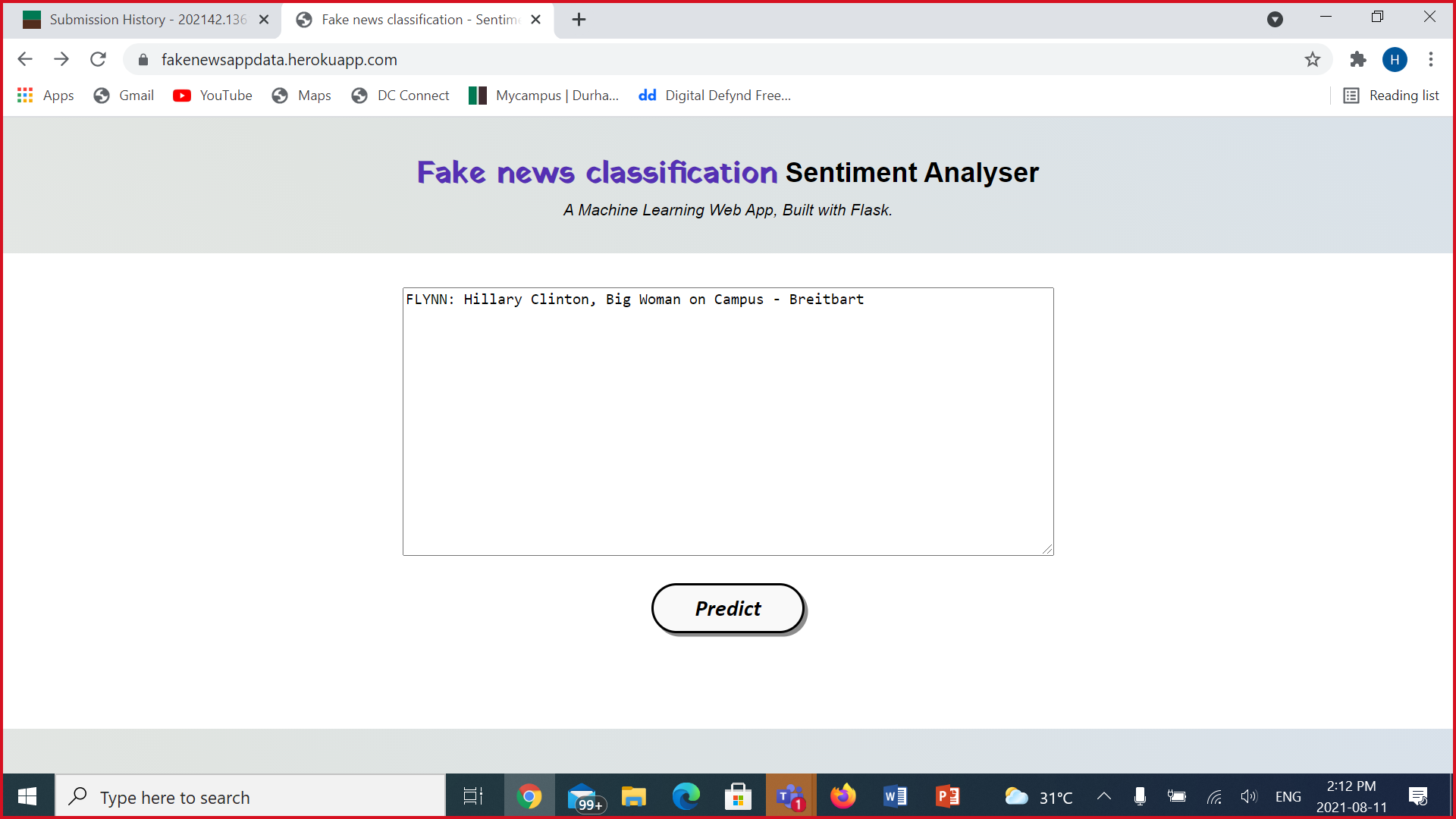
Using logistic Regression we got the highest accuracy score out of all the three models @ 93.68% and hence, we consider Logistic regression to be implemented and deployed on the server.

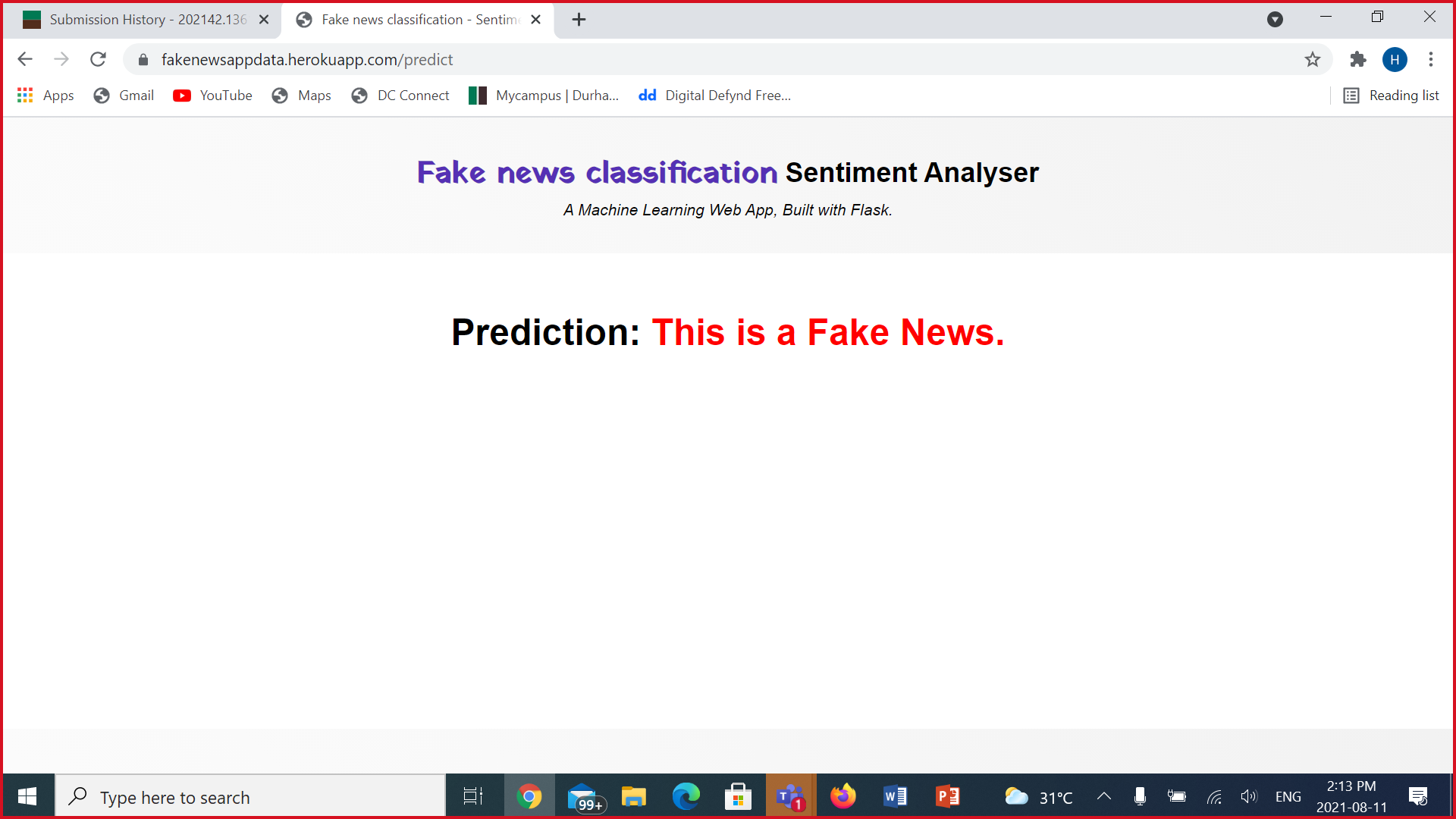
Accuracy Score:

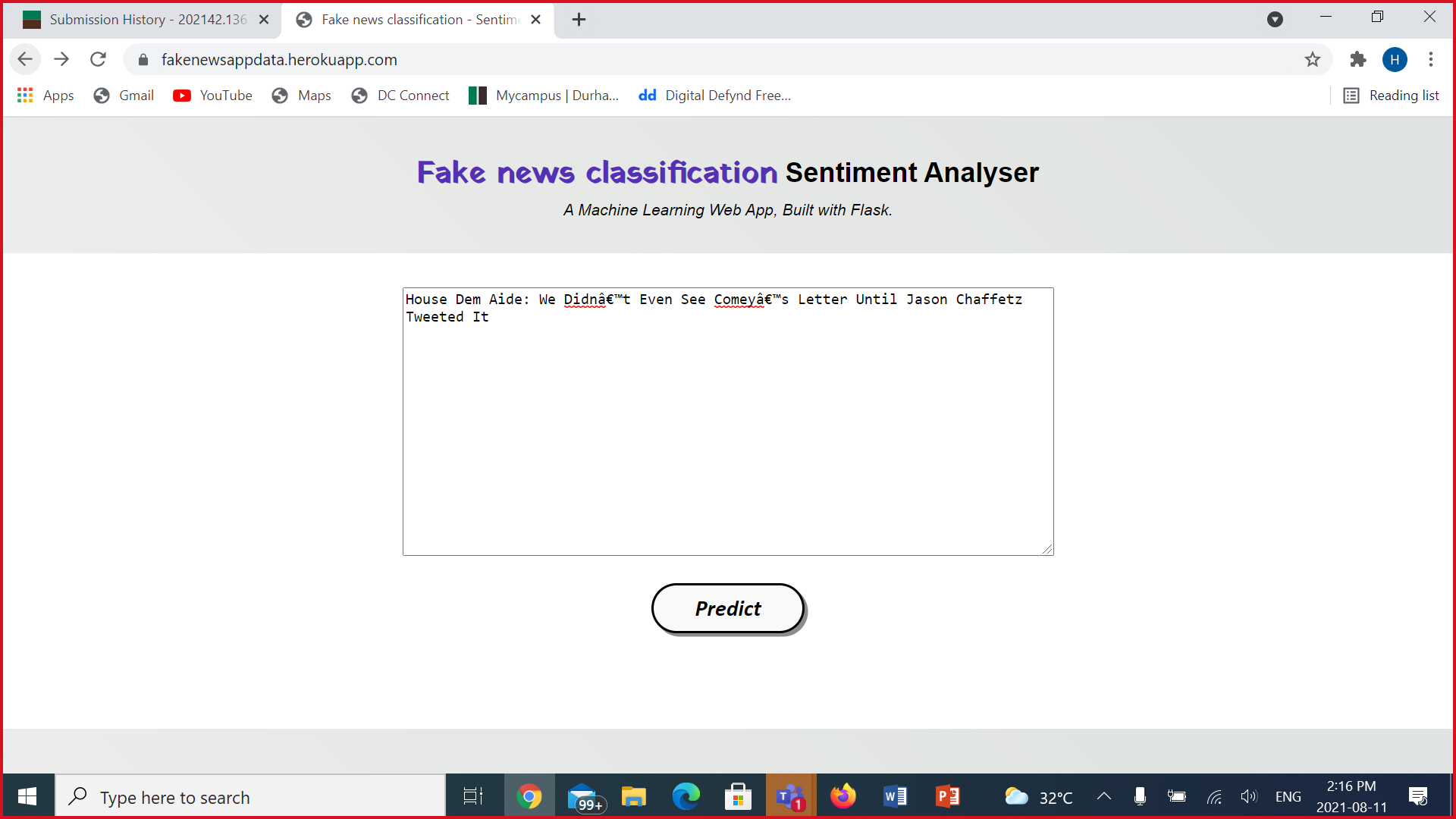
|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| Naïve Bayes | 90.89% |
| Random Forest Classification | 93.49% |
| Logistic Regression | 93.68% |

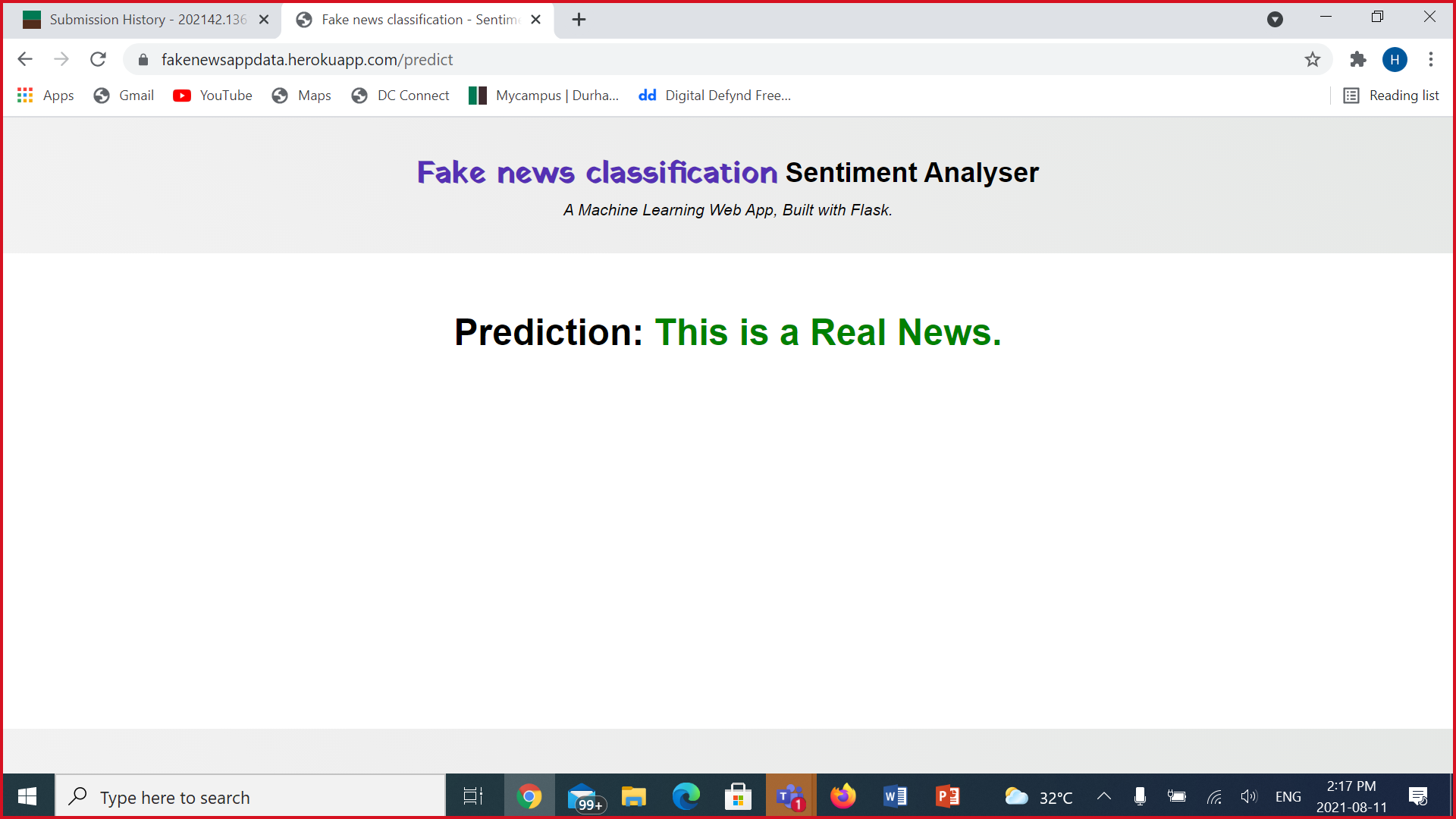
Demo:

[*https://fakenewsappdata.herokuapp.com/*](https://fakenewsappdata.herokuapp.com/)

Attached Screenshots demonstrates an example of Fake News



Attached Screenshots demonstrates an example of Real News



Time Plan:

Agile Methodologies:

We would be dividing the whole project into weekly sprints and follow up at the very end of the sprint for any requirement change or backlog. Every sprint will be divided into different stories (tasks) which each member can work on independently or dependently according to the task assigned.

Our main objective is to complete the Fake news classification system in around 4 Sprints as our final deliverable date is 12 August

Sprint 1: Week 1 - 12/7/21 to 16/7/21

* Our main objective will be to complete the data collection and data preprocessing

Sprint 2: Week 2 – 12/7/21 to 16/7/21

* Exploratory Data Analysis
* Modelling Approach Discussion

Sprint 3: Week 3 – 26/7/21 to 30/7/21

* Modelling
* Evaluation

Sprint 4: Week 4 – 2/8/21 to 6/8/21

* Deployment of the system

