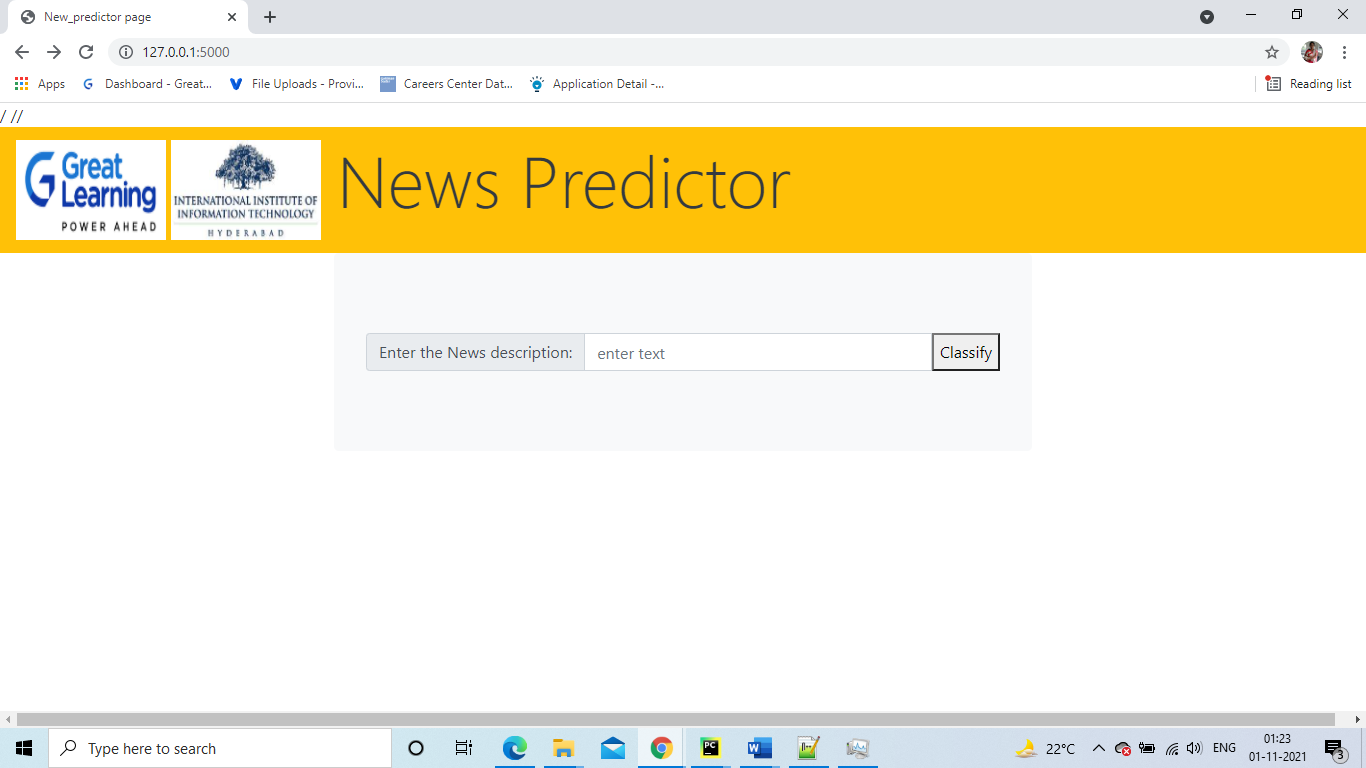
**NEWS ARTICLE CLASSIFIER**

**By**

**Swati Surampudi**

**and**

**Oggu Rachana**



**Table of Contents**

[**1. Introduction**](#_30j0zll)2

[1.1 Requirements](#_tyjcwt) 3

[**2. Architecture Diagram**](#_1t3h5sf) 4

[**3. Technical Documentation**](#_4d34og8) 5

[3.1 Data Ingestion](#_2s8eyo1) 5

[3.2 Data Preparation](#_uqilwyobgp4n) 9

[3.3 Data separation and Model training](#_mzwapfmcufzh) 10

[3.4 Model Deployment](#_6y5w1qdjea62) 11

# 1. Introduction

Classify News Articles into categories - With information overload today users are inundated with news articles of all topics, even the ones which may not be relevant to users. Design a system which can classify incoming news articles and appropriately tag the corresponding category. Develop a data pipeline which includes the all the following stages of Machine Learning Project Life Cycle :

1.Data Ingestion

2.Data Preparation

3.Data segregation & Model Training

4.Model Deployment

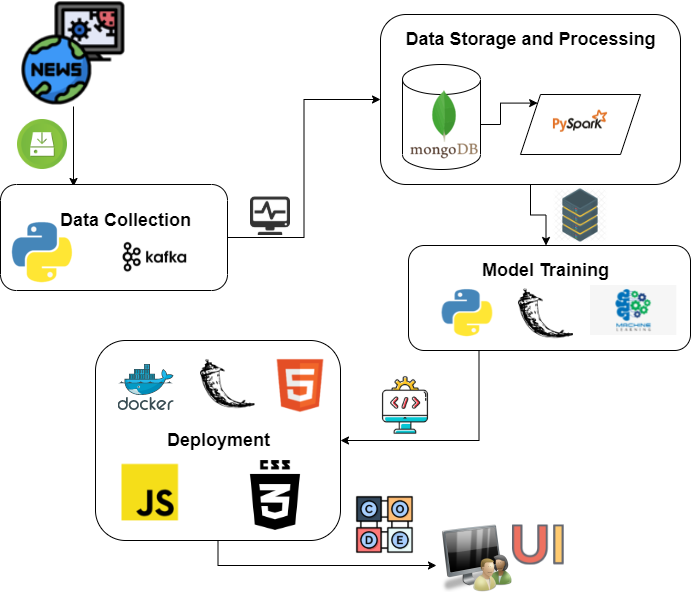
5.Model Prediction

## **1.1 Requirements**

For the implementation of News classifier the following are the requirements :

* Python IDE- VS Code
* MongoDB as database storage
* Pyspark for stream processing
* POSTMAN for testing Flask API’s
* Apache Zookeeper + Kafka for message streams
* Tensorboard for monitoring the progress of model training
* MLFLOW for model versioning +hyper-parameters versioning
* Python cookiecutter templates for setting up project

# 2. Architecture Diagram

ss

# 3. Technical Documentation

## **3.1 Data Ingestion**

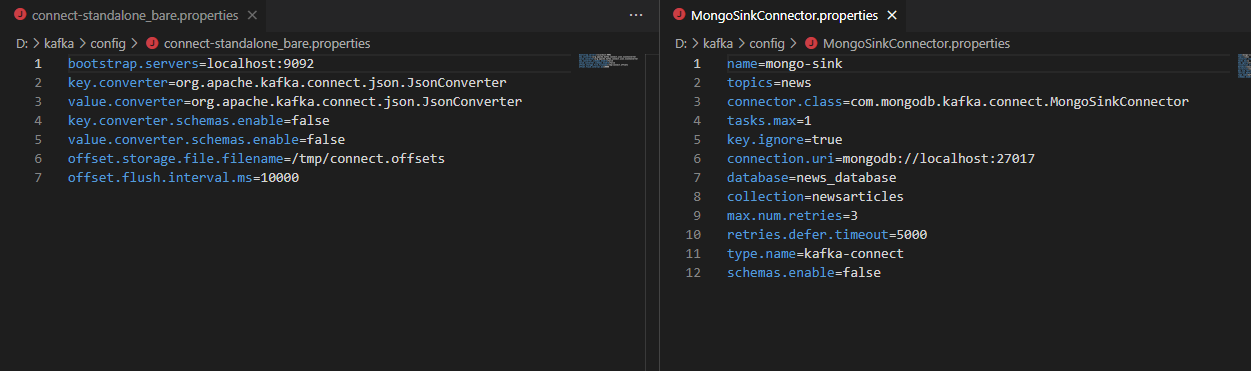
For the data ingestion collected the data from the API: mediastack.com

We are using Kafka for streaming data from the API and collecting it in MongoDB. Below are steps followed for data ingestion.

**Kafka Installation:**

To run Kafka and Zookeeper in the local systems below need to be installed :

* Installed Java 8 version and set the user variables and system variables Path.
* Installed Kafka from <https://archive.apache.org/dist/kafka/2.8.1/kafka_2.13-2.8.1.tgz>
* Configure MongoSinkConnector and stand alone bare properties in the kafka/config by changing the parameter values as mentioned in the screenshot.



* Downloaded the jar file from the link below and placed it in the path kafka/lib.

<https://repo1.maven.org/maven2/org/mongodb/kafka/mongo-kafka-connect/1.6.1/>

**MonoDB Installation**:

* Installed Community edition of MongoDB using link below:

<https://www.mongodb.com/try/download/community>

**Python Installation**:

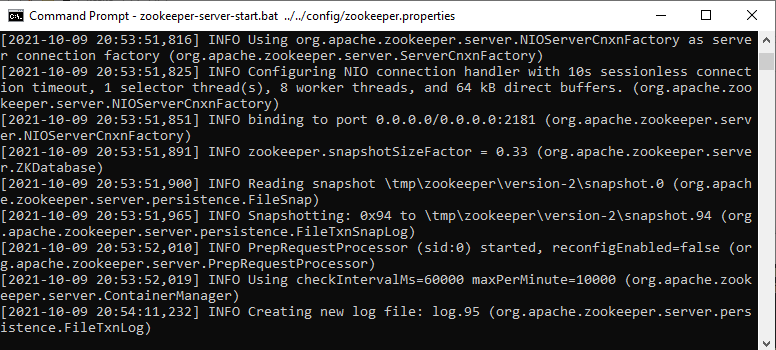
* pip install kafka-python

Ran the below commands after changing the Path:cd D:\kafka\bin\windows

**Terminal-1:**

Start the Zookeeper in a terminal

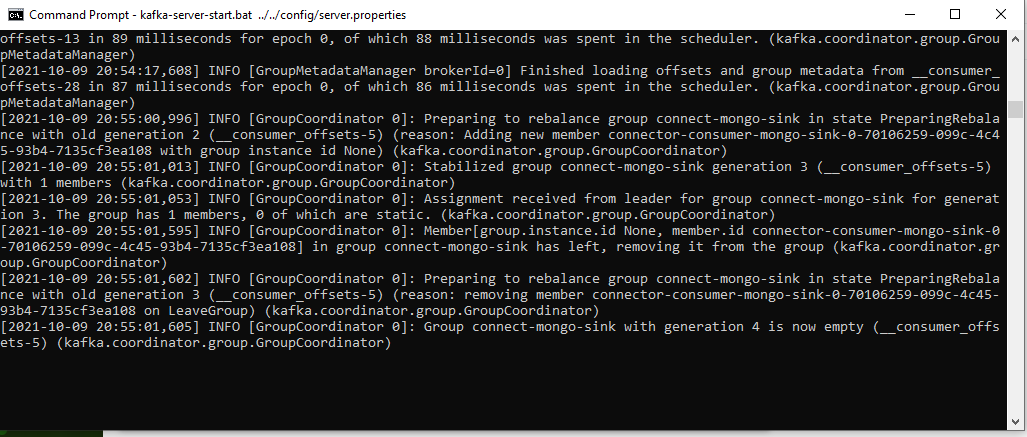
**zookeeper-server-start.bat ../../config/zookeeper.properties**



**Terminal-2:**

Start the kafka-server in another terminal

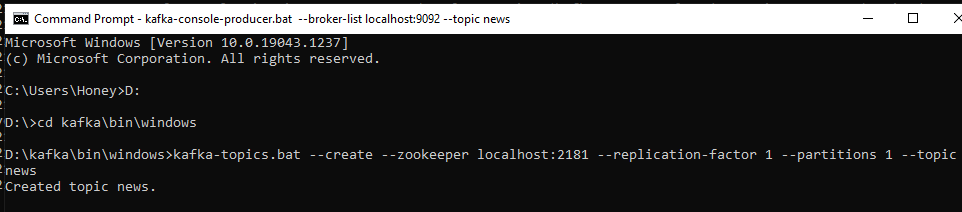
**kafka-server-start.bat ../../config/server.properties**



**Terminal-3:**

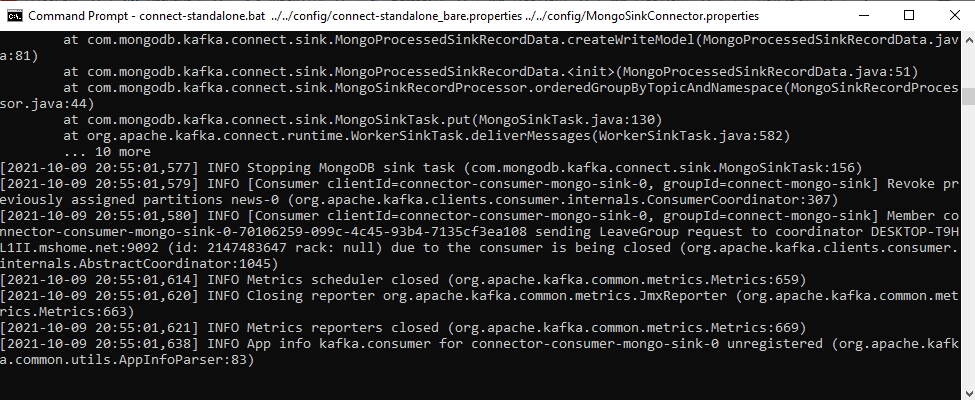
create the news topic in a new terminal

**kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic news**



In the same terminal run the mongodb connector command

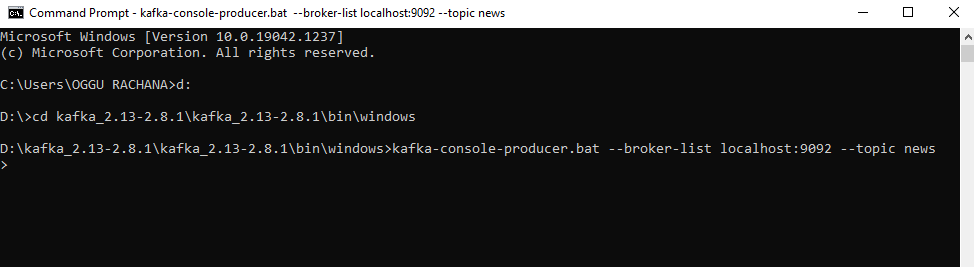
**connect-standalone.bat ../../config/connect-standalone\_bare.properties ../../config/MongoSinkConnector.properties**



**Terminal-4:**

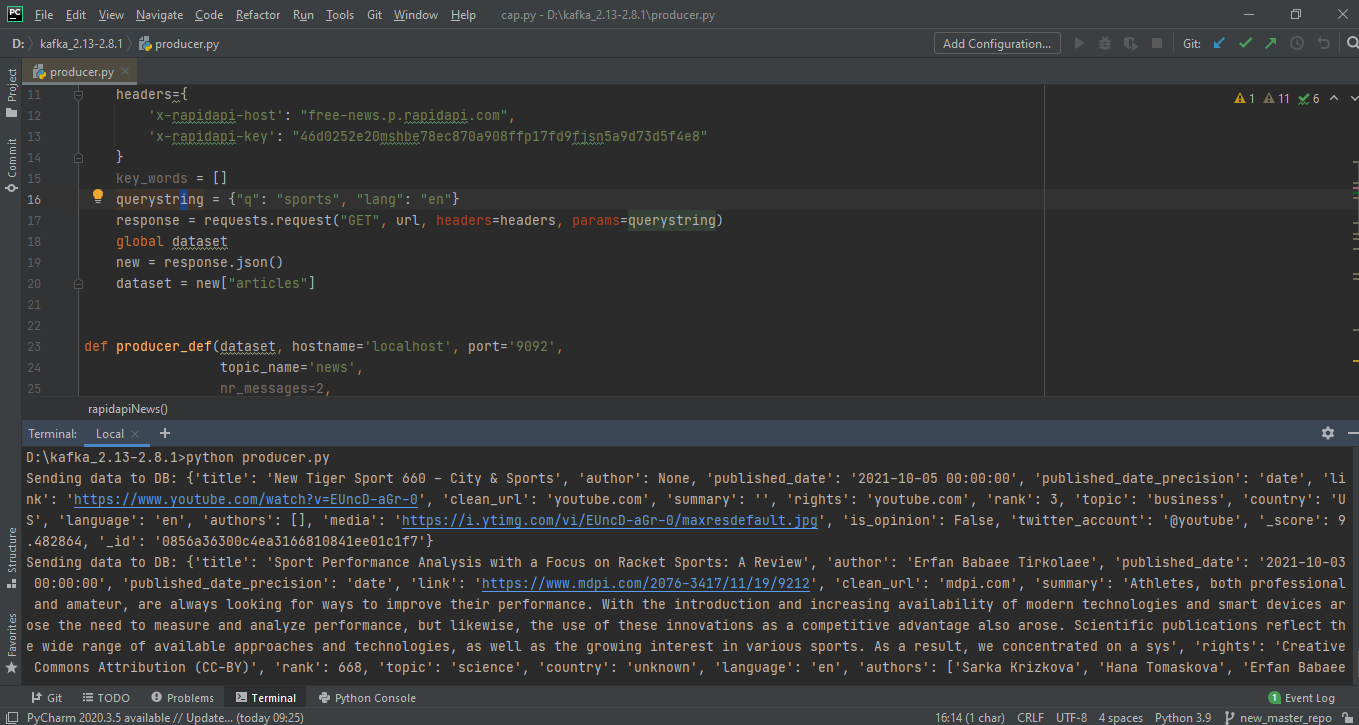
Creating Kafka Producer

**kafka-console-producer.bat --broker-list localhost:9092 --topic news**

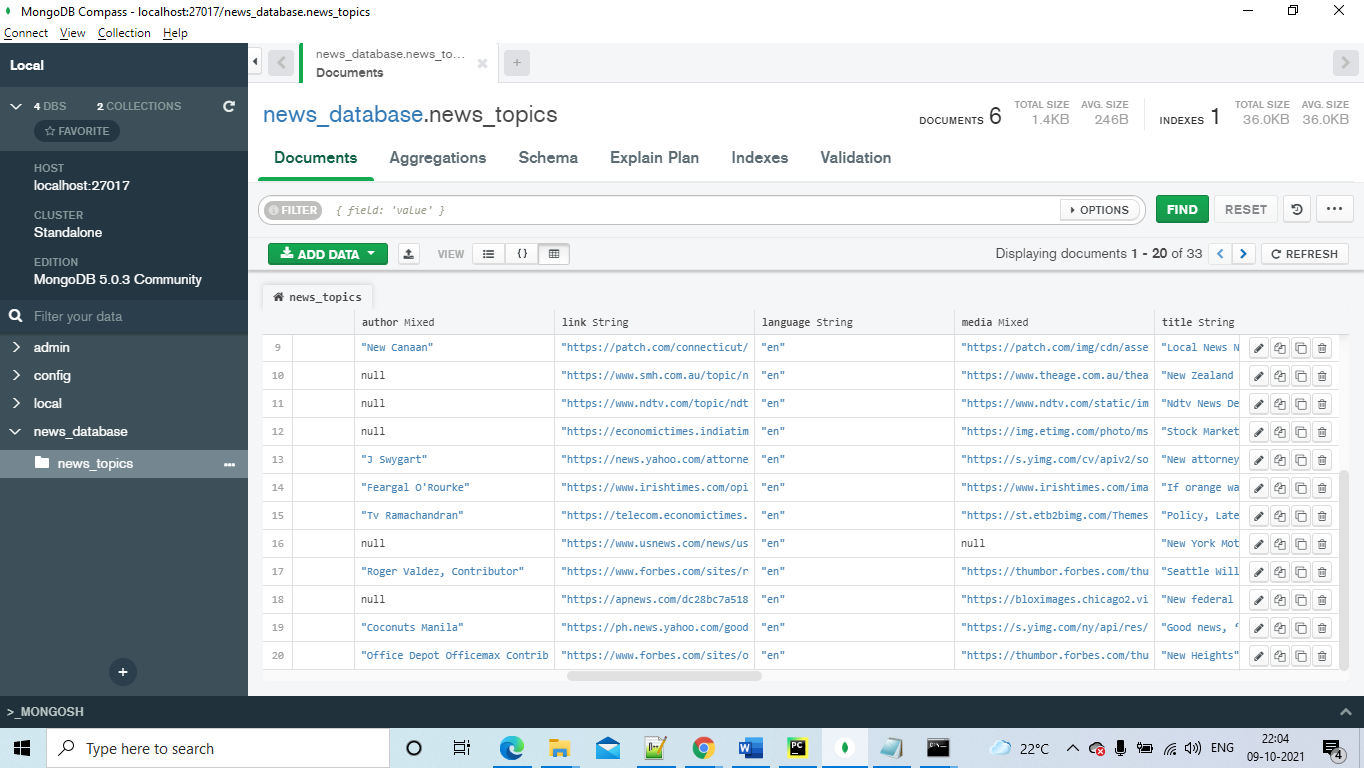
****

**Producer:**

Run the prodecur\_week1.py file which inserts the data into mongoDB



Data inserted into MongoDB



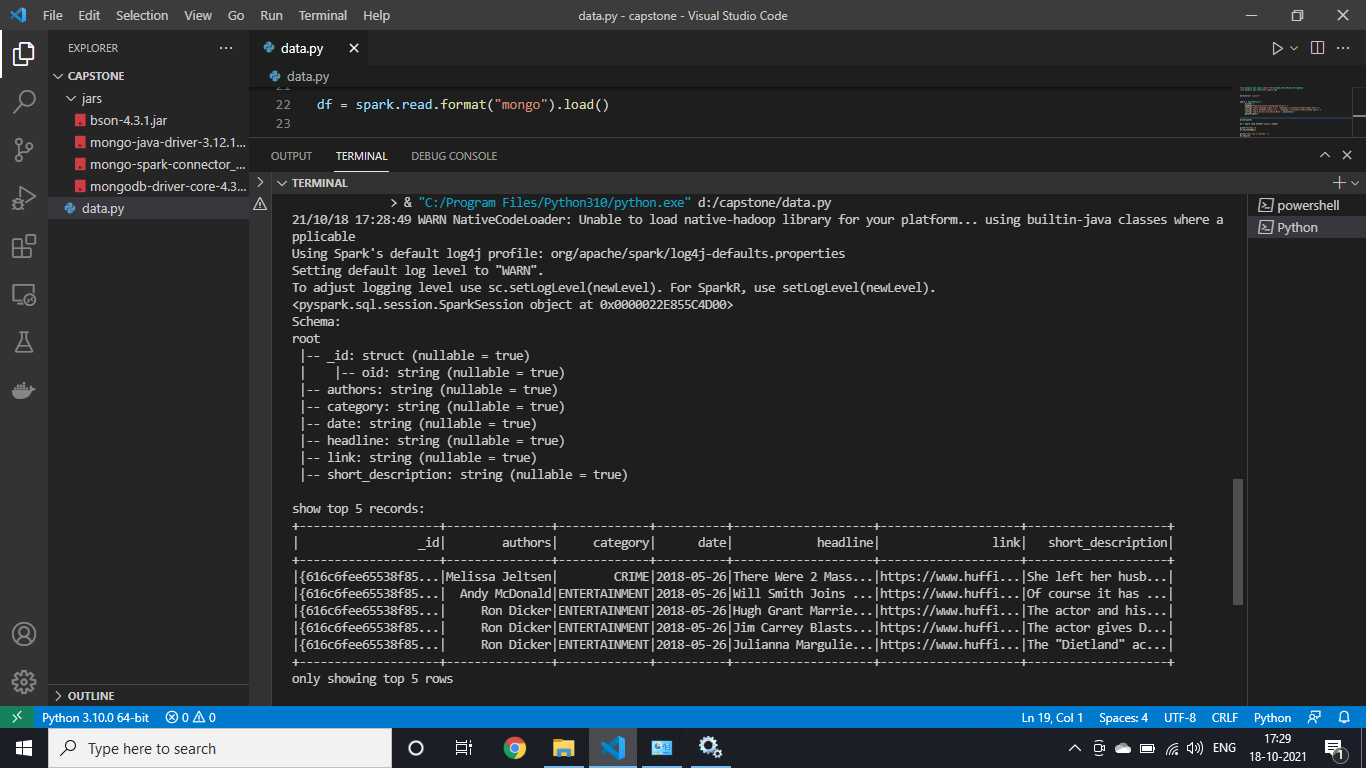
Collected the raw data into the database.

## **3.2 Data Preparation**

Once we get the raw data into the database, we have to clean the data like check for nulls, unwanted columns etc.

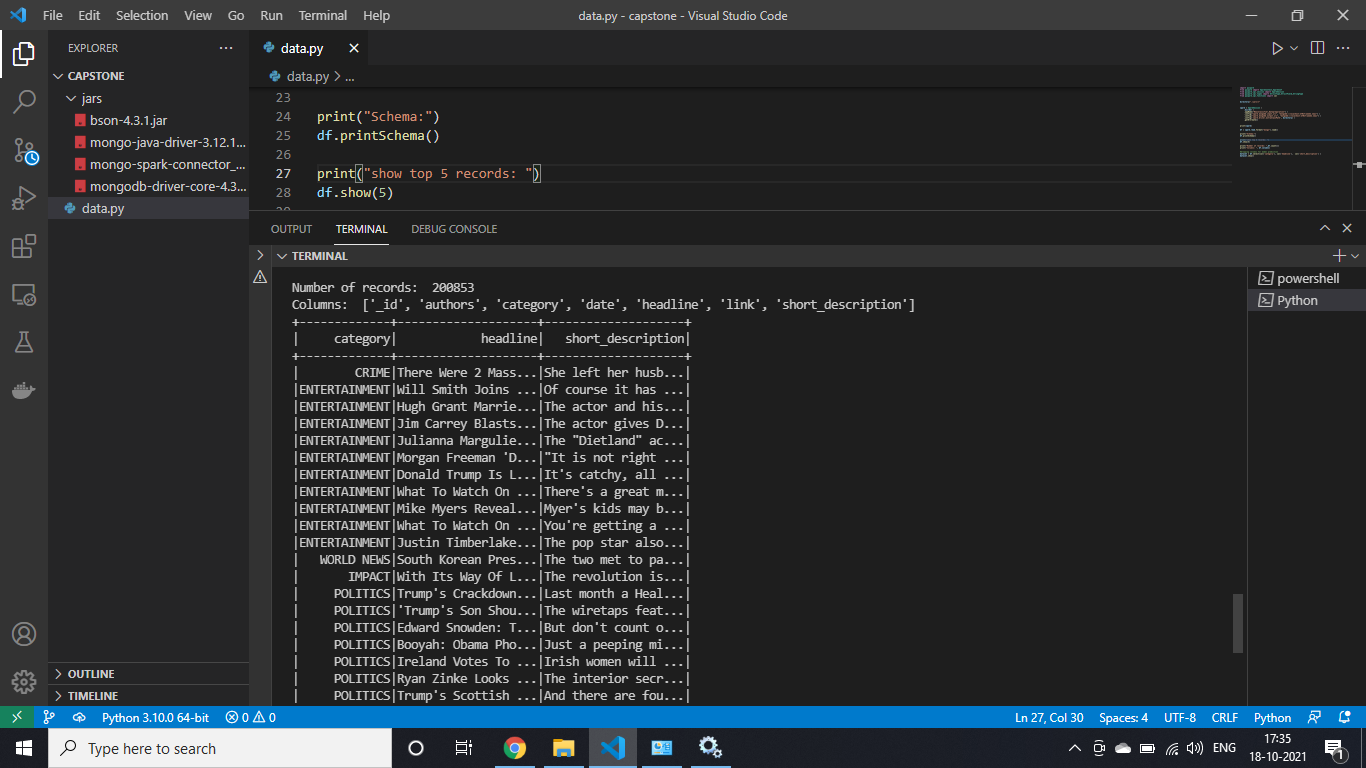
We are using pyspark for cleaning the data. For pyspark to connect with MongoDB, we need a mongo spark connector to be installed. (mongo-spark-connector\_2.12-3.0.1)

That connector is used for reading raw data from the database.



Then we remove the unwanted columns, nulls in the data and keep only required columns and replace nulls with default values.

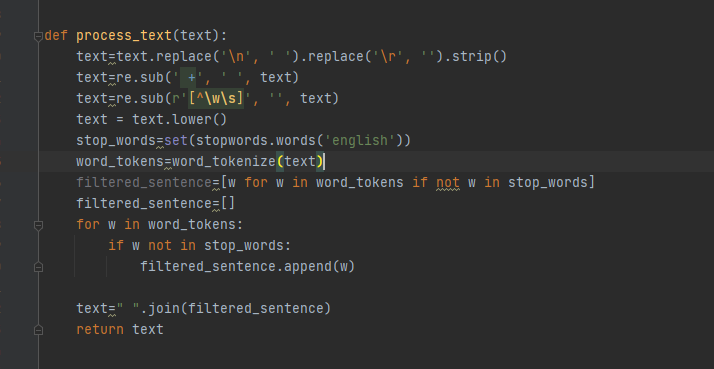
Once data is cleaned, exported the data to .csv file



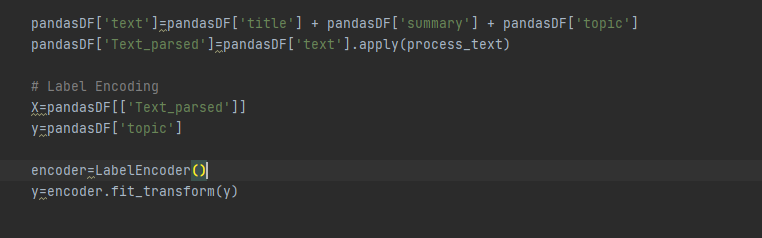
The cleaned data is inserted into the database for creation of the model.

## **3.3 Data separation and Model training**

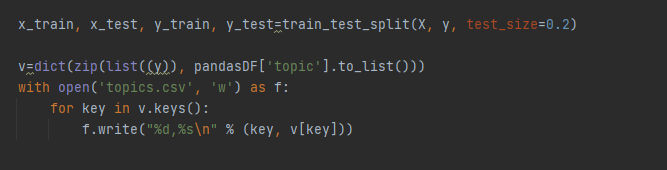
**Data Processing:** The data is processed further to remove any special characters, spaces,symbols etc from the text.



**Data labelling**: Defining x,y and label encoding Y data.



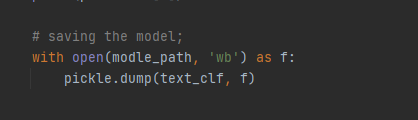
**Test and Train:** Defining test and train data and creating a .csv file which contains the list of the encoded Y column data.



Trained the model using the MultinomialNB classifier and performed the test.

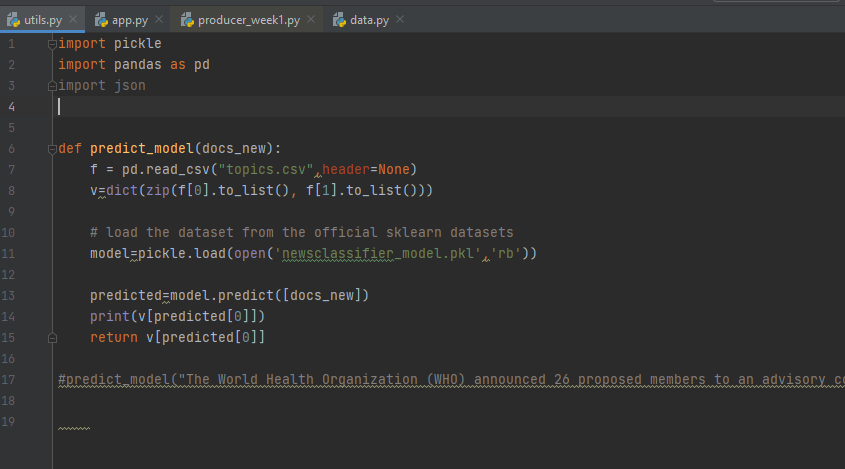


**Model Save:** The model has trained with the data available in the database. FInally saving the model into the pkl file.



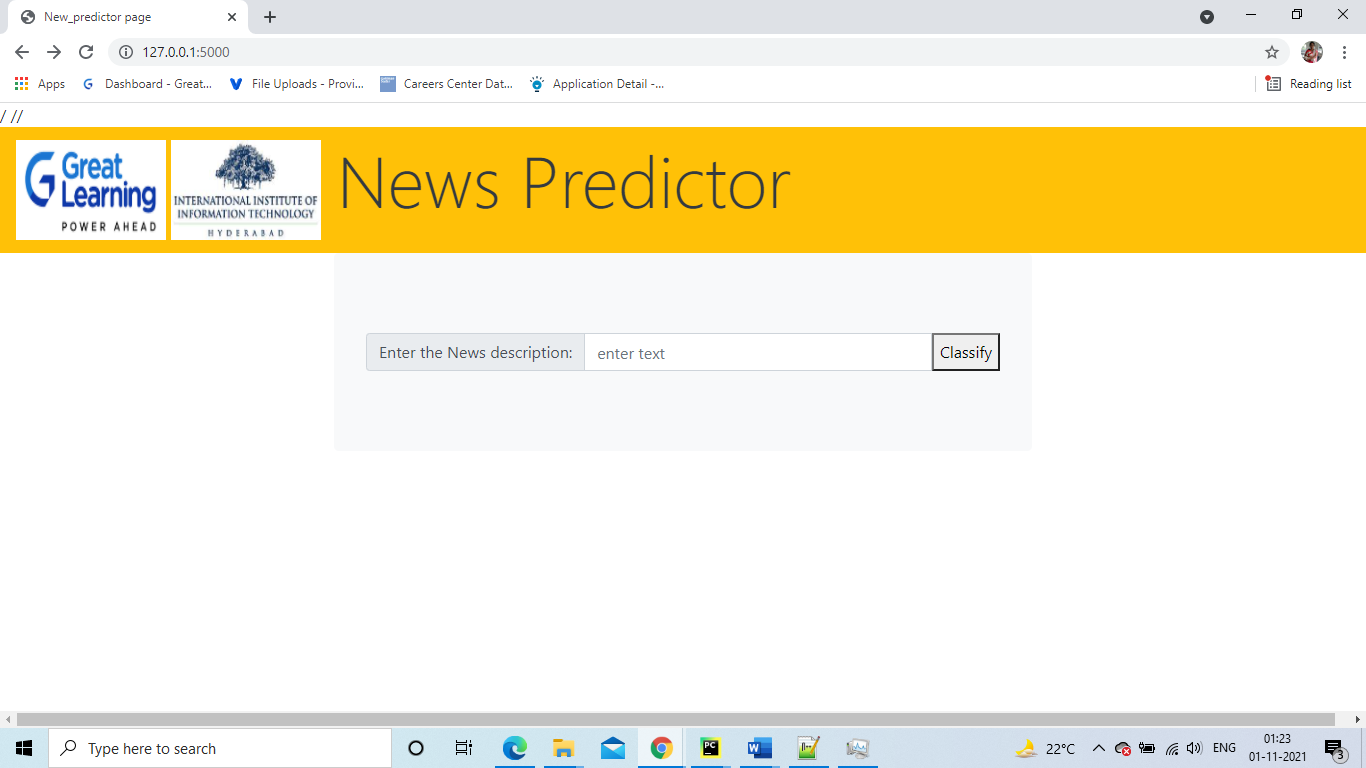
## **3.4 Model Deployment**

Created utils.py files which loads data from pkl file from above and predicts the data given as input.



Using flask API for deployment of the code.

**Final Web application:**



Used docker to contain the web application and submit for final submission.