**NEWS ARTICLES CLASSIFIER**

**Document History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rev #** | **Date** | **Author** | **Verified By** | **Brief description of Change** |
| 1 |  |  |  |  |

**Table of Contents**

[1 Introduction 3](#_Toc84704570)

[2 Project Specifications 4](#_Toc84704571)

[2.1 Requirements 4](#_Toc84704572)

[2.2 Architecture Diagram 4](#_Toc84704573)

[3 Technical Documentation 5](#_Toc84704574)

[Installation & Process 5](#_Toc84704575)

[#3.1 Data Ingestion 6](#_Toc84704576)

# 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

# Project Specifications

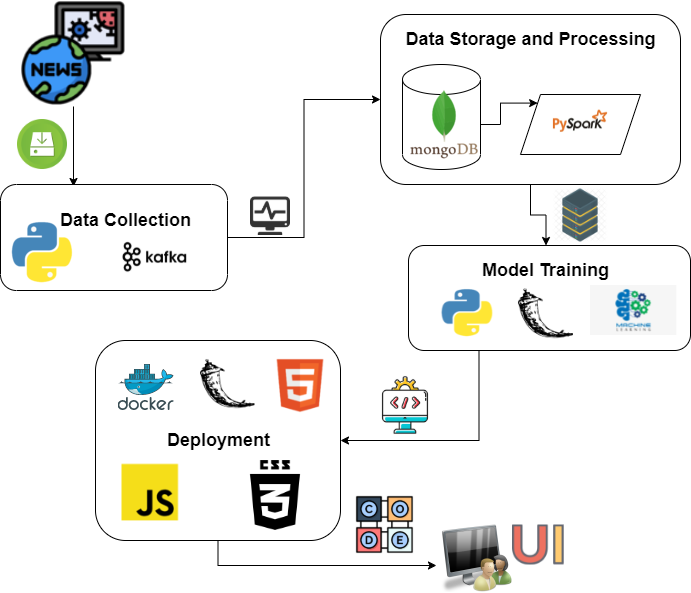
## 

### 2.1 Requirements

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

* Python IDE- VS Code
* Virtual Environment
* 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

### 2.2 Architecture Diagram

ss

# Technical Documentation

## Installation & Process

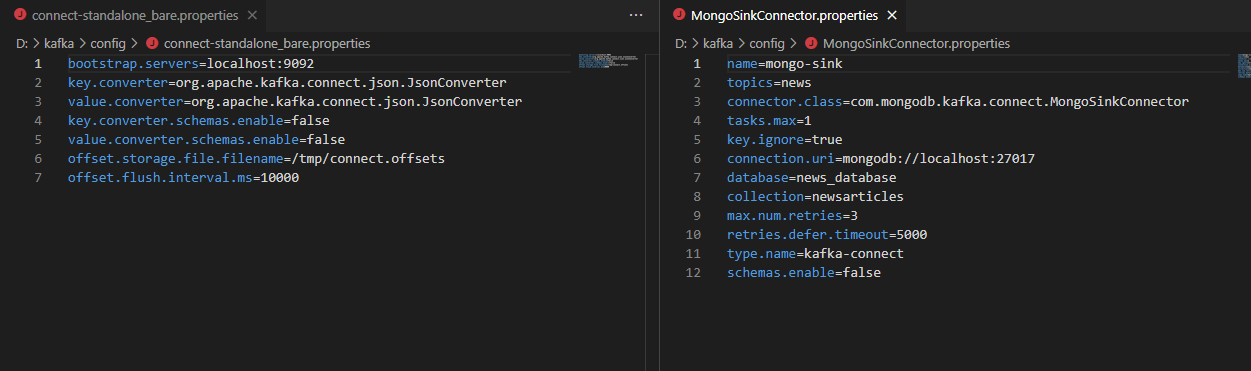
### #3.1 Data Ingestion

Install Tooling:

Kafka:

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>
* Configured 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 below link and placed it in the path kafka/lib.

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

MonoDB:

Installed Community edition of Mongo DB using link below:

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

Python:

* Installed pip pacakges requests, json,kafkapython.

Steps for fetching Data:

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

* ter-1: zookeeper-server-start.bat ../../config/zookeeper.properties
* ter-2: kafka-server-start.bat ../../config/server.properties
* ter-3:

1. kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic news
2. connect-standalone.bat ../../config/connect-standalone\_bare.properties ../../config/MongoSinkConnector.properties

* ter-4: kafka-console-producer.bat --broker-list localhost:9092 --topic news

### #3.2 Data preparation

Install Tooling:

* Downloaded the jar file from the below link and placed it in folder containing pyspark file.

https://repo1.maven.org/maven2/org/mongodb/spark/mongo-spark-connector\_2.12/3.0.1/

* <https://repo1.maven.org/maven2/org/mongodb/spark/mongo-spark-connector_2.12/3.0.1/mongo-spark-connector_2.12-3.0.1.jar>.
* https://repo1.maven.org/maven2/org/mongodb/bson/

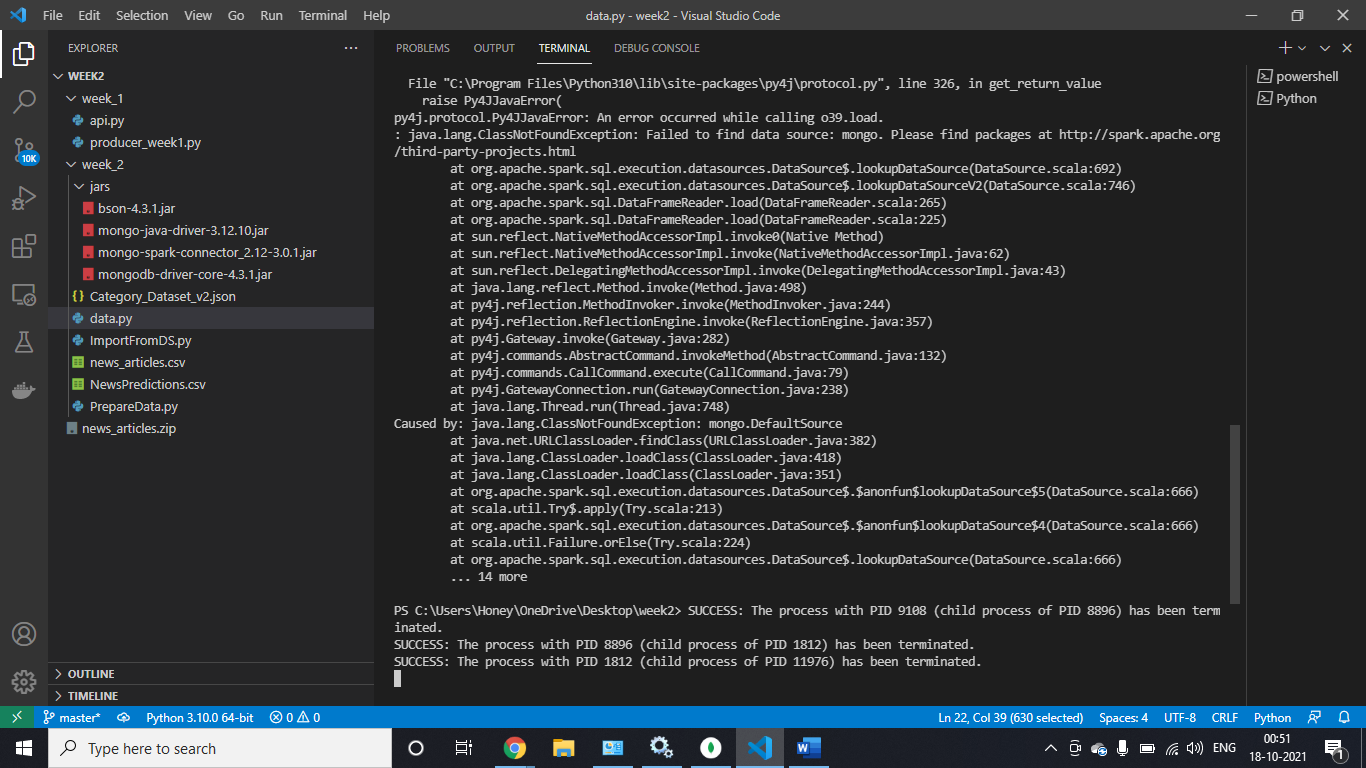
Source data:

<https://mediastack.com/documentations>

challenges:

* py4j.protocol.Py4JJavaError: An error occurred while calling o39.load.

: java.lang.ClassNotFoundException: Failed to find data source: mongo.



* py4j.protocol.Py4JJavaError:

An error occurred while calling o39.load. : java.lang.NoClassDefFoundError: com/mongodb/connection/DefaultClusterFactory

Jars:

<https://repo1.maven.org/maven2/org/mongodb/mongo-java-driver/3.12.10/mongo-java-driver-3.12.10.jar> <https://repo1.maven.org/maven2/org/mongodb/mongodb-driver-core/4.3.1/mongodb-driver-core-4.3.1.jar> <https://repo1.maven.org/maven2/org/mongodb/bson/4.3.1/bson-4.3.1.jar>