### **FLOOD MONITORING SYSTEM**

###### 622621121018: HARIGOPINATH.V

**INTRODUCTION :**

##### A flood monitoring system is used to monitor a rise in water levels. The system comprises sensors that are deployed in cities or any area of interest. The sensors can be connected to either the main electricity or can be solar-powered. These sensors are deployed on bridges, wells, lakes, or beaches to measure water levels in real-time and continuously send data remotely to the centralized data system management via different networks such as GSM, mobile cell networks, or Wi-Fi.roduction:

##### 

**NODE-RED:**

Node-RED is a powerful tool for building Internet of Things (IoT) applications with a focus on simplifying the ‘wiring together’ of code blocks to carry out tasks. It uses a visual programming approach that allows

developers to connect predefined code blocks, known as ‘nodes’, together to perform a task. The connected nodes, usually a combination of input nodes, processing nodes and output nodes, when wired together, make **up a ‘flow’.**

**Mit app inventor:**

**# I'm not going to read in the packages & data again since it's**

**# already in our current environment.**

**def create\_keywordProcessor(list\_of\_terms, remove\_stopwords=True,**

**custom\_stopword\_list=[""]):**

**""" Creates a new flashtext KeywordProcessor and opetionally**

**does some lightweight text cleaning to remove stopwords, including**

**any provided by the user.**

**"""**

**# create a KeywordProcessor**

**keyword\_processor = KeywordProcessor()**

**keyword\_processor.add\_keywords\_from\_list(list\_of\_terms)**

**# remove English stopwords if requested**

**if remove\_stopwords == True:**

**keyword\_processor.remove\_keywords\_from\_list(stopwords.words('english'))**

**# remove custom stopwords**

**keyword\_processor.remove\_keywords\_from\_list(custom\_stopword\_list)**

**return(keyword\_processor)**

**def apply\_keywordProcessor(keywordProcessor, text, span\_info=True):**

**""" Applies an existing keywordProcessor to a given piece of text.**

**Will return spans by default.**

**"""**

**keywords\_found = keywordProcessor.extract\_keywords(text, span\_info=span\_info)**

**return(keywords\_found)**

**# create a keywordProcessor of python packages**

**py\_package\_keywordProcessor = create\_keywordProcessor(list\_of\_packages,**

**custom\_stopword\_list=["kaggle", "http"])**

**# apply it to some sample posts (with apply\_keywordProcessor function, omitting**

**# span information)**

**for post in sample\_posts:**

**text = apply\_keywordProcessor(py\_package\_keywordProcessor, post, span\_info=False)**

**print(text)**

**['dataset', 'dataset']**

**['dataset', 'public', 'dataset', 'dataset', 'dataset', 'dataset', 'dataset', 'common', 'events']**

**['html', 'html', 'html', 'html']**

**IMPORTANT PICKLE:**

**# save our file (make sure our file permissions are "wb",**

**# which will let us \_w\_rite a \_b\_inary file)**

**pickle.dump(py\_package\_keywordProcessor, open("processor.pkl", "wb"))**

**# check our current directory to make sure it saved**

**!ls**

**\_\_notebook\_\_.ipynb \_\_output\_\_.json processor.pkl**

**[10/23, 6:42 PM] Naveen: # read in a processor from our pickled file. Don't forget to**

**# include "rb", which lets us \_r\_ead a \_b\_inary file.**

**pickle\_keywordProcessor = pickle.load(open("processor.pkl", "rb"))**

**# apply it to some sample text to make sure it works**

**apply\_keywordProcessor(pickle\_keywordProcessor, "I like pandas numpy and seaborn")**

**[('pandas', 7, 13), ('numpy', 14, 19), ('seaborn', 24, 31)]**