DIGITAL VILLAGE HACKATHON

Binary Coders Presents- FarmCast

By Yash Awasthi, Aamna Umar, Vikramjit Singh and Antrix Chadha

INTRODUCTION

Our Problem Statement- Develop a smart App for PA (precision agriculture) to generate irrigation schedule recommendations based on real-time weather and short-term forecasted data to better meet the plant water needs of a given period, conserving water while also minimizing nutrient leaching from the root zone due to excessive irrigation.

Why We Choose This Problem.

We have been curious to develop a one stop solution for farmers that wander around about putting their human knowledge to build up a crop and even a slight weather prediction failure causes huge effects on their yields.

So we thought to put up the power of Machine Learning and Deep Learning in use and with the help of Long short-term memory (LSTM) which is an artificial recurrent neural network (RNN) architecture used in the field of deep learning to predict out the NDVI graphs of a particular crop, and with that data, farmers are able to calculate upon how to manage their crop.

Architecture

For this hackathon we've built the Algorithm part and we also attempted to design the front-end bit of the dashboard website that the urer(farmer) will use to interact with the hackathon.

1. **Algorithm Part**

   We used this RNN architecture to design the algorithm with which we predicted the some-bit of the Graph of the NDVI of the same crop data that we plotted fully(separately).

2. **Front-End UI/UX Design Part**

   We tried to keep the front-end part of the website quite aesthetic and minimalist with all the features of scheduling and watching out on the future effects on the crop using the algo in interactive ways by using graphs and also by putting suggestion channels for the user to grow crops according to location and weather conditions.

Brief Information about the algorithm

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection systems). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.

Recurrent networks … have an internal state that can represent context information. … [they] keep information about past inputs for an amount of time that is not fixed a priori, but rather depends on its weights and on the input data. … A recurrent network whose inputs are not fixed but rather constitute an input sequence can be used to transform an input sequence into an output sequence while taking into account contextual information in a flexible way.

Brief Information about NDVI

The NDVI is computed as the difference between near-infrared (NIR) and red (RED) reflectance divided by their sum.

$$\mathrm{NDVI}_i = \frac{\mathrm{NIR}-\mathrm{RED}}{\mathrm{NIR}+\mathrm{RED}}$$

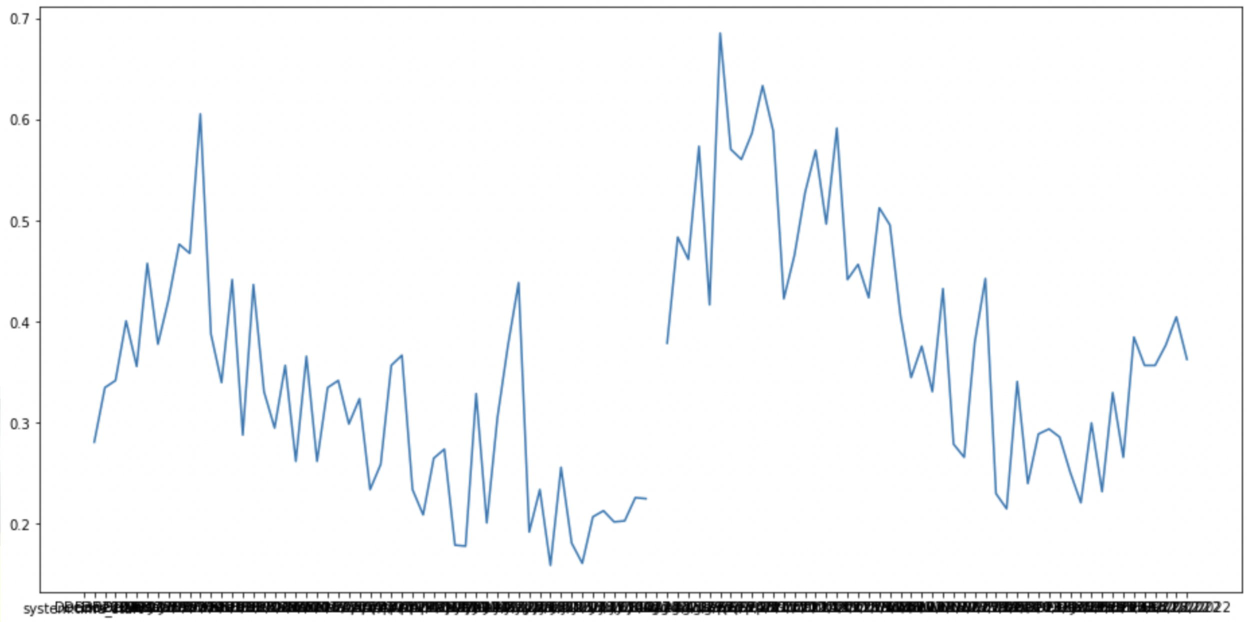
NDVIi represents smoothed NDVI (sNDVI) observed at time step i and their ratio yields a measure of photosynthetic activity within values between − 1 and 1. Low NDVI values indicate moisture-stressed vegetation and higher values indicate a higher density of green vegetation. It is also used for drought monitoring and famine early warning (Wardlow et al., 2007; Javadnia et al., 2009). For computing NDVI anomaly (z-score) which was to require a series of images with historical year [long-term averages (LTAs)] for each period (in our case: month) in the year. The anomaly (z-score) indicators of vegetation condition can be calculated as Z_{NDVI} and its value is widely used for monitoring vegetation anomalies (Klisch and Atzberger, 2016). It is calculated at pixel level from the sNDVI data as.

$$Z_{\mathrm{NDVI}} = \frac{\mathrm{NDVI}_i - \mathrm{NDVI}_{\mathrm{mean},m}}{\sigma_{\mathrm{NDVI}}}$$

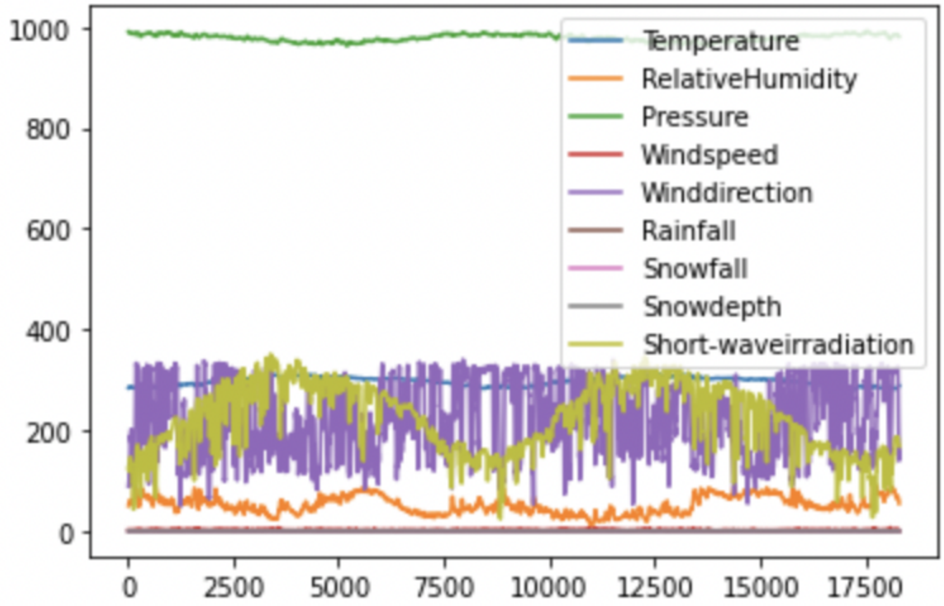
where Z_{NDVI} is the standard difference (z-score) of NDVI at time step i, NDVIi = sNDVI is the observed at time step i, is the monthly mean sNDVI values, and σNDVI is the standard deviation of sNDVI values at month (m) respectively. σNDVI indicates the signed number of standard deviations is above or below the mean.

Results that we achieved:

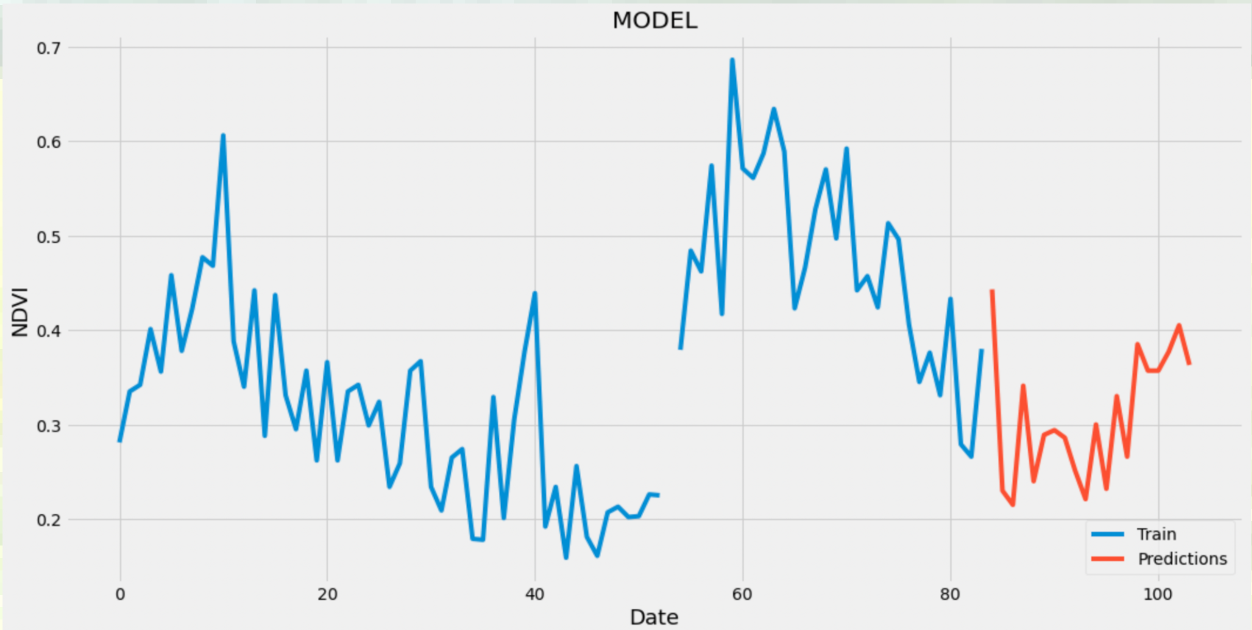
Graph of Dates VS NDVI (Fully Informed)(Algorithm Unused):-



Graph of Locational-Conditions that are used to calculate NDVI:



Graph when the model was used to predict the end bit of the NDVI graph:



We have made the front-end part bit on Figma...The Link for the same is

https://www.figma.com/file/nT0cD9ntG0QKs9P5WWbDec/Digital-Village?node-id=2%3A4