A PROJECT REPORT

for

CROP PRICE PREDICTION

using Machine Learning

By

Harish Inavolu

CONTENTS:

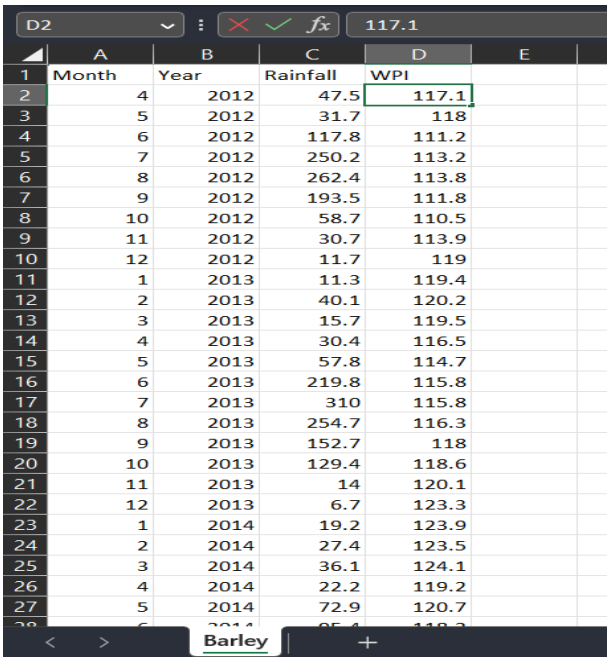
ABSTRACT

In recent years, due to ensure climatic variations, natural calamities, and other issues, the prices of crops have changed to a huge level. Farmers are uninformed of these concerns, which results in a loss. The Crop Price Forecasting System is a well-designed system that provides reliable outcomes in the determination of crop price. Machine Learning with the prediction model has gained popularity through its promising results. This project also incorporates its application, which involves exploring several regression models for predicting agricultural production. The crop price prediction assists the farmers to plant their next crop to be grown and avoid hyperinflation. The dataset has 23 different crops altogether. The effectiveness of several models has been compared, and the best-fit model with the necessary accuracy has emerged.

DATA DESCRIPTION:

The agricultural data from the public repository are collected to create the dataset. We were able to get the data that describes the specifics of the rainfall for each crop. The table below displays a sample of the collected data set along with some of its properties. WPI stands for Wholesale Price Index. The model is developed using verified datasets from https://data.gov.in/ Following data collection, the dataset is used to run the algorithms and forecast prices for 23 commodities over a period of time of about 12 months (future 12-month prices).

Data Gathering: The dataset contains Annual Rainfall and WPI from April 2012 to December 2018 for 23 different crops namely - Pigeon pea, Pearl millet, Barley, Cotton, Chickpea, Peanut, Sorghum, Jute, Maize, Red lentil, Greengram, Niger Seed, Rice, Finger millet, Safflower, Sesame, Soybean, Sugarcane, Sunflower, Black Gram and Wheat.



	A	B	C	D
1	Month	Year	Rainfall	WPI
2	4	2012	47.5	117.1
3	5	2012	31.7	118
4	6	2012	117.8	111.2
5	7	2012	250.2	113.2
6	8	2012	262.4	113.8
7	9	2012	193.5	111.8
8	10	2012	58.7	110.5
9	11	2012	30.7	113.9
10	12	2012	11.7	119
11	1	2013	11.3	119.4
12	2	2013	40.1	120.2
13	3	2013	15.7	119.5
14	4	2013	30.4	116.5
15	5	2013	57.8	114.7
16	6	2013	219.8	115.8
17	7	2013	310	115.8
18	8	2013	254.7	116.3
19	9	2013	152.7	118
20	10	2013	129.4	118.6
21	11	2013	14	120.1
22	12	2013	6.7	123.3
23	1	2014	19.2	123.9
24	2	2014	27.4	123.5
25	3	2014	36.1	124.1
26	4	2014	22.2	119.2
27	5	2014	72.9	120.7

Fig: data gathered for barley crop

Data Cleaning: Data cleaning is a mandatory process that one should use while dealing with machine learning. Using the dataset, we conducted several statistical analyses and data visualization techniques to determine which data-cleansing procedures were required. Before moving on to the sophisticated techniques we can apply to every dataset in a machine-learning project, there were some quite basic data-cleansing processes. To verify that all the data types are valid, we removed all the null and missing values which are the basic anomalies present in the dataset.

Data Exploration: Exploratory data analysis (EDA) is a prominent phase in data research, query, and investigation that involves summarizing significant characteristics of various datasets using data visualization techniques. We utilized EDA to identify repeated trends, anomalies, and test theories, which enabled us to draw conclusions and determine the best approach for monitoring data sources with greater precision.

The previous system predicted the cropping rate for 1 year and that includes the graphical representation of specific crops for the previous year and the upcoming year. Today, numerous researchers have put various models into practice to forecast agricultural yields. But because the techniques were used incorrectly, these models contain a number of flaws. The User Interface of the above-proposed system is not user-friendly. It can only provide predicted data for 1-month

PROJECT DESIGN:

We utilized the Python package index (PPI), specifically the Flask package, to deploy the project. An instance for Flask was created, acting as a central configuration object that sets up the necessary components of the application. While Flask does not have built-in database facilities, we used a package to connect the user interface (UI) to a SQL database. The UI was developed using HTML/CSS and displays crop details, including price fluctuations over time, which are displayed through a graph. The Flask framework connects the UI to the model. We developed the interface using Visual Studio Code and implemented machine learning models such as Decision Tree Regressor and Random Forest Regressor with the help of Python libraries like numpy, pandas, and sklearn.

The steps of machine learning prediction are as follows:
Step 1: Creating a dataset with the info on rainfall and the wholesale price index.
Step 2: Choose "X" as an independent variable from the aforementioned dataset, then all rows and columns starting at positions 1, 2, and 3.
Step 3: Choose "y" in column 4 of the dataset to select the dependent variable.
Step 4: To train the Decision Tree Regression/Random Forest Regression model, just use training data.
Step 5: Using the decision tree regressor and the random forest regressor, try to fit the x and y variables.
Update the UI (User Interface) using the projected values in step six.

Two machine learning algorithms were used in this project: Decision tree regressor and Random forest. The Decision tree regressor constructs regression and classification models in the form of a tree diagram by breaking down a dataset into smaller sub-sets. The resulting decision tree consists of nodes and leaves that represent tested attribute values, making it suitable for handling categorical and numerical data. With no backtracking, a greedy top-down search across the space of potential branches is carried out using the ID3 algorithm. The other algorithm, named Random Forest, uses bagging as an element of ensemble methods and is a supervised learning algorithm. To get reliable predictions, the ensemble combines predictions from multiple decision trees. With such useful changes, random forest is used as a meta-estimator to combine various decision trees.

This resulted in the creation of 600 decision trees, each of which yielded a forecast. After that, an average is computed based on all the predictions. The random forest predictor then receives the data and generates an output for us. The ensemble performs this so that we can obtain precise accuracy.

Accuracy

To find a better prediction algorithm from Decision Tree Regressor and Random Forest Regressor, we used a KFold cross-validation algorithm.

K Fold cross-validation

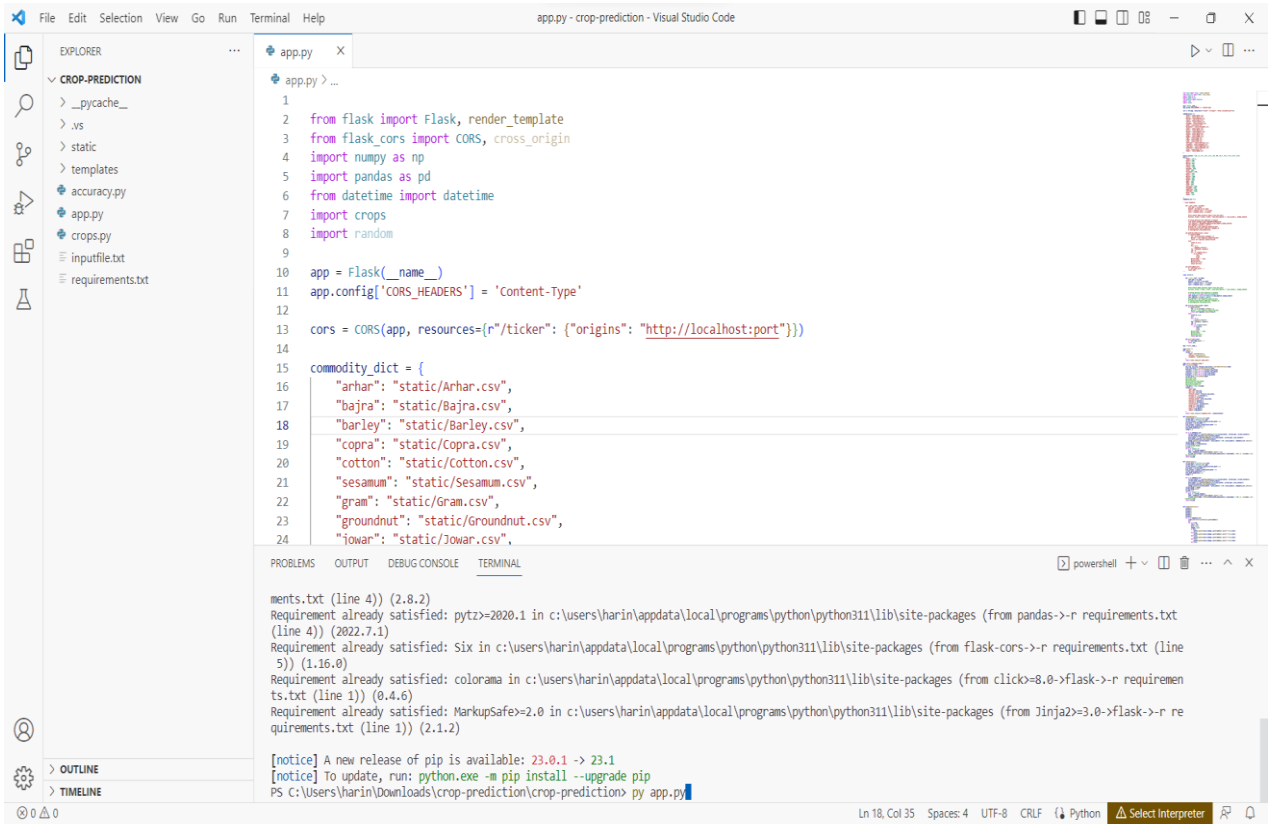
A resampling technique called cross-validation is used to assess machine learning models on a small data sample. The process contains a single parameter, k, that designates how many groups should be created from a given data sample. When a particular number for k is selected, it may be substituted for k in the model's reference, such as when k=10 is used to refer to cross-validation by a 10-fold factor.

In applied machine learning, cross-validation is mostly used to gauge how well a machine-learning model performs on untrained data. That is, to use a smaller sample to gauge how the model will generally perform when used to make predictions on data that was not included during the model's training.

The general procedure is as follows:

- Randomly shuffle the dataset.
- Group the dataset into k equal parts.
- Evaluate the model's scores and fit a model to each distinct group.
- Use a sample of the model evaluation score to summarize the model's score upon evaluation.

To determine accuracy, we used a k-fold cross-validation algorithm and performed 10 splits for each dataset. We then calculated the average of all splits for each dataset and stored the values in separate lists. Finally, we calculated the overall average by taking the average of all the accuracy scores obtained from the 23 datasets.


File Edit Selection View Go Run Terminal Help
app.py - crop-prediction - Visual Studio Code
EXPLORER
CROP-PREDICTION
__pycache__
.vs
static
templates
accuracy.py
app.py
crops.py
inputfile.txt
requirements.txt
app.py
app.py > ...
from flask import Flask, render_template
from flask_cors import CORS, cross_origin
import numpy as np
import pandas as pd
from datetime import datetime
import crops
import random
app = Flask(__name__)
app.config['CORS_HEADERS'] = 'Content-Type'
cors = CORS(app, resources={r"/ticker": {"origins": "http://localhost:port"}})
commodity_dict = {
"arhar": "static/Arhar.csv",
"bajra": "static/Bajra.csv",
"barley": "static/Barley.csv",
"copra": "static/Copra.csv",
"cotton": "static/Cotton.csv",
"sesamum": "static/Sesamum.csv",
"gram": "static/Gram.csv",
"groundnut": "static/Groundnut.csv",
"jowar": "static/Jowar.csv",
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
ments.txt (line 4)) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\harin\appdata\local\programs\python\python311\lib\site-packages (from pandas->-r requirements.txt
(line 4)) (2022.7.1)
Requirement already satisfied: Six in c:\users\harin\appdata\local\programs\python\python311\lib\site-packages (from flask-cors->-r requirements.txt (line
5)) (1.16.0)
Requirement already satisfied: colorama in c:\users\harin\appdata\local\programs\python\python311\lib\site-packages (from click>=8.0->flask->-r requiremen
ts.txt (line 1)) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\harin\appdata\local\programs\python\python311\lib\site-packages (from Jinja2>=3.0->flask->-r re
quirements.txt (line 1)) (2.1.2)
[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\harin\Downloads\crop-prediction\crop-prediction> py app.py
OUTLINE
TIMELINE
Ln 18, Col 35 Spaces: 4 UTF-8 CRLF Python Select Interpreter


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
py
[notice] A new release of pip is available: 23.0.1 -> 23.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\harin\Downloads\crop-prediction\crop-prediction> py app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 140-498-431

PROJECT MILESTONES:

- Based on the existing system, we developed a UI that generates the price of crops for the next 12-month period.
- We achieved an accuracy of 93.1 % by using Random Forest Regressor which is higher compared to the other model used.
- The UI developed by us is more interactive and user-friendly as compared to the previous model.
- The K-fold cross-validation technique utilized in our project is an incremental feature that folds the data into different k parts depending upon the input given (k).
- The model forecasts the crop price variation trends for the next 12 months and also visualizes these trends in the form of a line graph

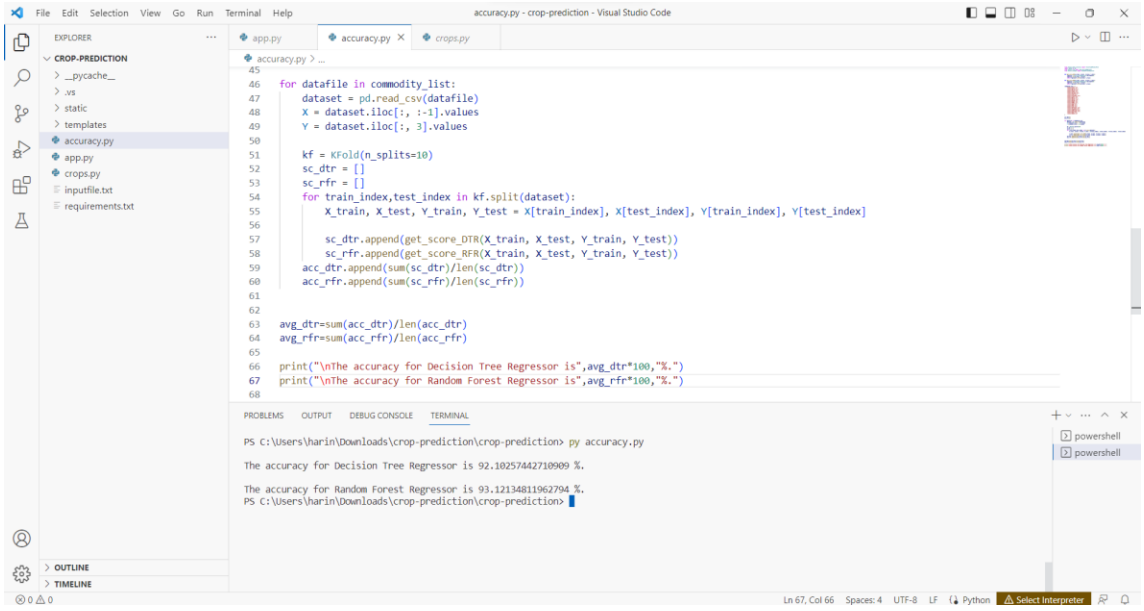
PROJECT RESULTS:

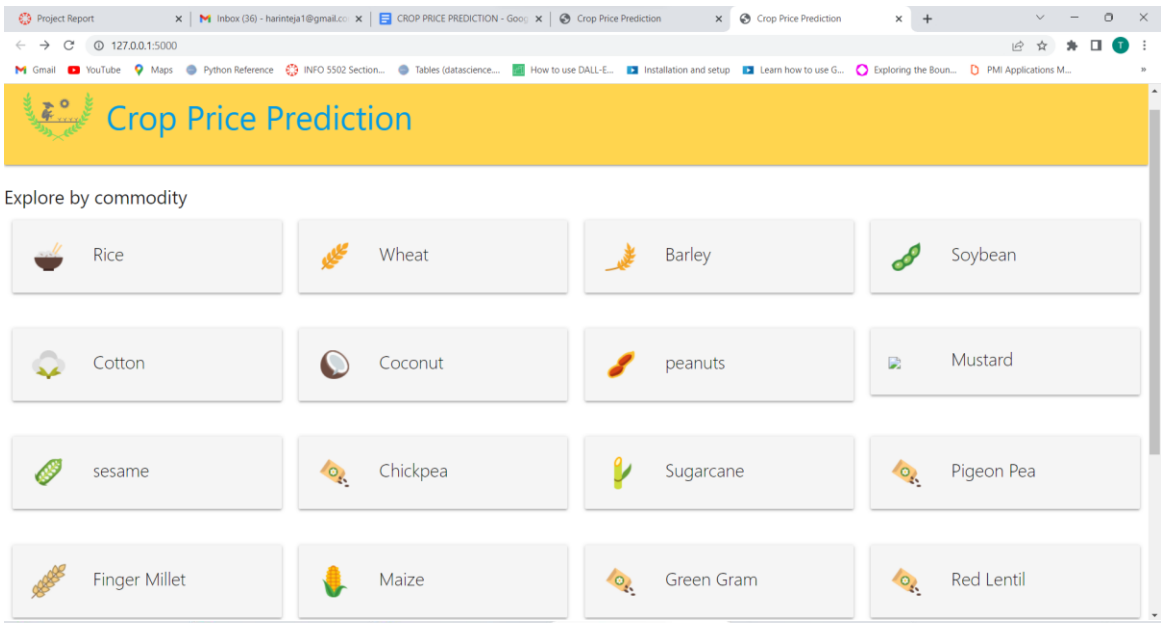
- Future changes to our project will assist in developing it to the extent feasible in each and every location (that used a GPS module), extracting the dataset for that location, and then using that dataset to boost accuracy and probability. Introducing the chat portals and making the application more user-friendly is expected to be another scope for enhancement. Farmers will be highly benefited from precise forecasts to raise profits.

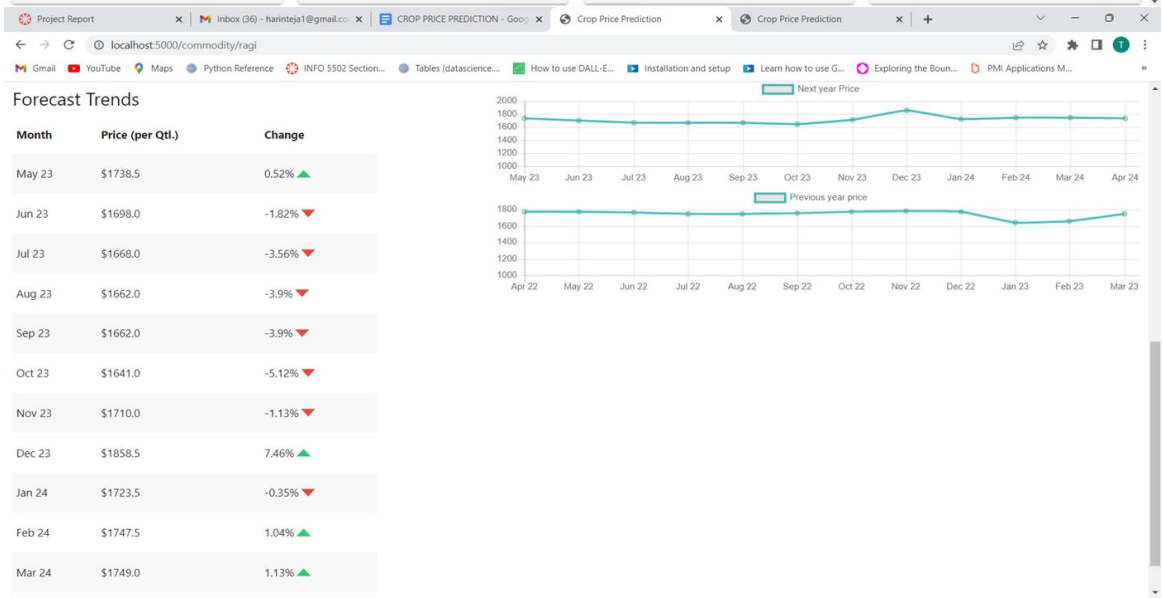
- On the basis of cutting-edge AI models, the model can be further trained with climate-aware farming practices, supplying the right fertilizers, and identifying methods of crop monitoring, and warning about insect outbreaks, and disease breakouts.

- The improvement of the application relies on making it available to most people, easier to use, and higher accuracy. The climate-aware farming techniques are also used to train the AI model.

- As a result, there can be a decrementation in the problems of the farmers and a hike in the income level. Several algorithms like decision trees, support vector machines, neural networks, deep learning, etc. can be used for crop price prediction.


accuracy.py - crop-prediction - Visual Studio Code
EXPLORER
CROP-PREDICTION
__pycache__
.vs
static
templates
accuracy.py
app.py
crops.py
inputfile.txt
requirements.txt
app.py
accuracy.py
crops.py
accuracy.py > ...
for datafile in commodity_list:
dataset = pd.read_csv(datafile)
X = dataset.iloc[:, :-1].values
Y = dataset.iloc[:, 3].values
kf = KFold(n_splits=10)
sc_dtr = []
sc_rfr = []
for train_index,test_index in kf.split(dataset):
X_train, X_test, Y_train, Y_test = X[train_index], X[test_index], Y[train_index], Y[test_index]
sc_dtr.append(get_score_DTR(X_train, X_test, Y_train, Y_test))
sc_rfr.append(get_score_RFR(X_train, X_test, Y_train, Y_test))
acc_dtr.append(sum(sc_dtr)/len(sc_dtr))
acc_rfr.append(sum(sc_rfr)/len(sc_rfr))
avg_dtr=sum(acc_dtr)/len(acc_dtr)
avg_rfr=sum(acc_rfr)/len(acc_rfr)
print("\nThe accuracy for Decision Tree Regressor is",avg_dtr*100,"%.")
print("\nThe accuracy for Random Forest Regressor is",avg_rfr*100,"%.")
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PS C:\Users\harin\Downloads\crop-prediction\crop-prediction> py accuracy.py
The accuracy for Decision Tree Regressor is 92.10257442710909 %.
The accuracy for Random Forest Regressor is 93.12134811962794 %.
PS C:\Users\harin\Downloads\crop-prediction\crop-prediction>
powershell
powershell
OUTLINE
TIMELINE
Ln 67, Col 66
Spaces: 4
UTF-8
LF
Python
Select Interpreter


Project Report
Inbox (36) - harinteja1@gmail.co
CROP PRICE PREDICTION - Goo
Crop Price Prediction
Crop Price Prediction
127.0.0.1:5000
Gmail
YouTube
Maps
Python Reference
INFO 5502 Section...
Tables (datascience...
How to use DALL-E...
Installation and setup
Learn how to use G...
Exploring the Boun...
PMI Applications M...
Crop Price Prediction
Explore by commodity
Rice
Wheat
Barley
Soybean
Cotton
Coconut
peanuts
Mustard
sesame
Chickpea
Sugarcane
Pigeon Pea
Finger Millet
Maize
Green Gram
Red Lentil


Project Report
Inbox (36) - harinteja1@gmail.co
CROP PRICE PREDICTION - Goo
Crop Price Prediction
Crop Price Prediction
localhost:5000/commodity/ragi
Gmail
YouTube
Maps
Python Reference
INFO 5502 Section...
Tables (datascience...
How to use DALL-E...
Installation and setup
Learn how to use G...
Exploring the Boun...
PMI Applications M...
Forecast Trends
Month
Price (per Qtl.)
Change
May 23
$1738.5
0.52%
Jun 23
$1698.0
-1.82%
Jul 23
$1668.0
-3.56%
Aug 23
$1662.0
-3.9%
Sep 23
$1662.0
-3.9%
Oct 23
$1641.0
-5.12%
Nov 23
$1710.0
-1.13%
Dec 23
$1858.5
7.46%
Jan 24
$1723.5
-0.35%
Feb 24
$1747.5
1.04%
Mar 24
$1749.0
1.13%
Next year Price
2000
1800
1600
1400
1200
1000
May 23
Jun 23
Jul 23
Aug 23
Sep 23
Oct 23
Nov 23
Dec 23
Jan 24
Feb 24
Mar 24
Apr 24
Previous year price
1800
1600
1400
1200
1000
Apr 22
May 22
Jun 22
Jul 22
Aug 22
Sep 22
Oct 22
Nov 22
Dec 22
Jan 23
Feb 23
Mar 23

REPOSITORY:

The resources, archives, and code altogether are gathered in this link to the file:
Link to the repository

CODE:

The link to the pdf which includes the project code:
5502_Project_code.pdf

REFERENCES:

[1] Crop Price prediction using Random Forest and Decision Tree Regression, S Brunda, Nimish L, Chiranthan S, Arbaaz Khan, Assistant Professor, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, Volume: 07 Issue: 09 | Sep 2020

[2] S. Veenadhari, Dr. Bharat Misra, Dr. CD Singh.2019. Machine learning approach for forecasting crop yield based on climatic parameters .978-1-4799-2352- 6/14/$31.00 ©2014 IEEE

[3] Takeshi Yoshida Noriyuki Murakami and Hiroyuki Tauiji.2017. Hybrid Machine Learning Approach to Automatic Plant Phenotyping in Smart Agriculture. 978-1- 5090-5888-4/16/$31.00 @IEEE 2016.

[4] "Crop price forecasting using machine learning techniques: A comparative study" by Rakesh Kumar, Arvind Dewangan, and S. S. Rathore
Link: https://www.sciencedirect.com/science/article/pii/S2405452617304168.