**Project Report**

**General Adversarial Network**

To predict/Impute production(Yeild/Area) :

**1. Exploratory Data Analysis(EDA)**

Visualize and analyze the data by plotting graphs, histograms, and charts representing density functions to see the dependence of the attributes for the yield/area using **pandas, matplotlib, and Seaborn.**

**Data visualization-**

**2. Data Preprocessing**

**A. Data Cleaning**

Missing data- deleted rows having NaN values in the yield/area, latitude, and longitude column.

**B. Data Reduction**

Dimensionality reduction-From EDA chose the columns that are relevant for the prediction of Yield/Area like Latitude, longitude, crop, season, amount of fertilizers, Season, and year.

**C.Data Transformation**

Transformed the original data into a table with longitude as columns, latitude as rows with Yield /area as value by applying crop type, season, and year as filters with the help of a pivot table in Excel

**3. Table-to-image conversion -**

1. NaN Yield/Area values will be generated which are filled with zeros.
2. Convertedtheyield/area values to an array and normalized the values between 0 and 1 using **NumPy**.
3. Created an empty image to store the tabular data
4. Filled the pixel values with an array and converted the tabular data to an image using the **Pillow** library in Python

**4.**  Generated images using the **GAN model**

Steps:

1. Imported required libraries - **NumPy, matplotlib, Keras, Tensorflow**.Also imported neural network layers from Keras for the model.
2. Loaded and reshaped around 3000 images so that they could be used as inputs for the GAN model
3. Loaded the images and reshaped the images so that they can be taken as input for the model
4. Defined a utility function to build the generator by

building the input layers - batch normalization,conv2D, and activation functions such as **ReLu, tanh** and

generating the output images and joining then using model=sequential ().

1. Defined a utility function to build the discriminator by building convolutional layers - **conv2D, BatchNormalization, LeakyReLU, dropout,** and joining then using **model=sequential ().**

to classify whether an image is real or fake. And then build the output layer using the sigmoid activation function and check the validity of the image.

1. Defined a utility function to display the generated images
2. Build the General Adversarial Network- build a generator that checks the validity of the images generated by the generator with the help of the discriminator which is compiled using **binary\_crossentropy loss and Adam optimizer.**

Defined the combined model of the generator and the discriminator using binary\_crossentropy loss and Adam optimizer.

8. Trained the network by adding random noises and using discriminator and generator loss function and displayed images with epoch around 15000 and display interval =2500

**Comparing different models**

| **MLR** | * Used ‘Season’,’ Latitude..167’,’ longitude’ as features and ‘yield/area as target variable’ * Applied MLR(Multiple Linear Regression) models from SKlearn for a particular crop(ex. Paddy) * Evaluated the model with   R squared: 7.21  Mean Absolute Error: 677.1167  Mean Square Error: 3955915.06  Root Mean Square Error: 1988.948   * Created a new column in the datasheet representing Euclidean distance between the equator and latitude and longitude as points of a place. * Used ‘Season’,’ Area’ as features, and ‘yield/area as target variable’ * Applied MLR(Multiple Linear Regression) models from SKlearn for a particular crop(ex. Paddy) * Evaluated the model using sklearn with   R squared: 2.35  Mean Absolute Error: 623.6006127258513  Mean Square Error: 3992479.1549933176  Root Mean Square Error: 1998.1189041179  %Error: 41.04537397367536 |
| --- | --- |
| **Mice** | * Created a new column in the datasheet representing Euclidean distance between the equator and latitude and longitude as points of a place. * Taking ‘Season’,’ Area’, and ‘yield/area and dropping other columns. * Randomly created missing values in yield/area * Applied MICE(Multivariate Imputation by Chained Equations) model for a particular crop(ex. Paddy). * Evaluated the model with   Mean Absolute Error: 36.31848254346495  Mean Square Error: 442958.4865940915  Root Mean Square Error: 665.551265188559  %Error: 2.3992110508685428e+16 |
| **KNN** | * Taking ‘Season’,’Latitude..167’,’longitude’ and ‘yield/area and dropping other columns. * Randomly created missing values in yield/area * Evaluated the model with   R squared: 3.82  Mean Absolute Error: 0.16235255961500078  Mean Square Error: 44.32598382772922  Root Mean Square Error: 6.657776192373038 |
|  |  |