Web Application to perform Analysis and Prediction of Maize Crop Yield with Fertilizer Recommendation / Intelligent Agriculture App

Submitted by Praagna Prasad

Introduction:

PROBLEM STATEMENT AND DEFINITION:

India is an agrarian economy and agriculture plays an important role in contributing to the country's GDP. Machine learning is an emerging research field in crop yield analysis. Yield prediction is a very important issue in agriculture. Any farmer is interested in knowing how much yield he is about to expect. In the past, yield prediction was performed by considering farmer's experience on a particular field and crop. Thus, modernization of agriculture is very important and thus will lead the farmers of our country towards profit. Data analytic is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software. Earlier yield prediction was performed by considering the farmer's experience on a particular field and crop. However, as the conditions change day by day very rapidly, farmers are forced to cultivate more and more crops. Being this as the current situation, many of them don't have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. Also, the farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditons

Objective of the Work:

- To use machine learning too Predict the crop yield
- To provide easy to use user interface
- To increase the accuracy of crop yield prediction
- To take the major factors for yield prediction into account
- To recommend fertilizers on the basis of desired yield.

Crop Selected: Maize

Dataset location: USA

Literature Survey:

Crop Yield Prediction Using Machine Learning

Authors- B.Manjula Josephine, K.Ruth Ramya, K.V.S.N Rama Rao, Swarna Kuchibhotla, P. Venkata Bala Kishore, S. Rahamathulla

In this paper published in the INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 02, FEBRUARY 2020 edition, they explore Intelligent Analytics in Agriculture by describing an approach to predict millet crop yield prediction, which is done by taking high dimensional datasets. Their Methodology adopted includes using Random Forest Classifier to obtain a 99.74% of accuracy in calculating the millet crop yield prediction by taking various input

fields like soil, min temp, max temp, humidity, rainfall, etc. This Millet Crop Yield Prediction provided great insight into the data of crops and will provide immense value to the farmers to identify the crop losses and understand their yield to cost ratio. This model helped us to understand the process to predict and find out the accuracy of millet crop yield for both Support Vector Machine (SVM) and Linear Regression (LR) out of which Linear Regression Model was implemented in our project.

EFFICIENT CROP YIELD PREDICTION USING MACHINE LEARNING ALGORITHMS

Authors - Arun Kumar, Naveen Kumar, Vishal Vats

In this research paper they have applied descriptive analytics in the agriculture production domain for sugarcane crop to find efficient crop yield estimation. The model is built combing three datasets: Soil dataset, Rainfall dataset, and Yield dataset which is combined to create the dataset for the crop prediction. Several supervised techniques have been applied to find the actual estimated cost and the accuracy of several techniques. In this paper, three supervised techniques are used like K-Nearest Neighbour, Support Vector Machine, and Least Squared Support Vector Machine. It is a comparative study which tells the accuracy of training proposed model and error rate. The accuracy of training model should be higher and error rate should be minimum. And the proposed model is able to give the actual cost of estimated crop yield and it is label like as LOW, MID, and HIGH.

Crop Prediction System using Machine Learning

Authors- Prof. D.S. Zingade, Omkar Buchade, Nilesh Mehta, Shubham Ghodekar, Chandan Mehta

The proposed project detailed in this paper provides a solution for Smart Agriculture by monitoring the agricultural field which can assist the farmers in increasing productivity to a great extent. Weather forecast data is obtained from IMD (Indian Metrological Department) contains parameters such as temperature and rainfall and soil parameters repository gives insight into which crops are suitable to be cultivated in a particular area. This work presents a system, in form of an android based application, which uses data analytics techniques in order to predict the most profitable crop in the current weather and soil conditions. The proposed system will integrate the data obtained from repository, weather department and by applying machine learning algorithm: Multiple Linear Regression, a prediction of most suitable crops according to current environmental conditions is made. This provides a farmer with variety of options of crops that can be cultivated. Thus, the project model develops a system by integrating data from various sources, data analytics, prediction analysis which can improve crop yield productivity and increase the profit margins of farmer helping them over a longer run.

PREDICTION OF CROP YIELD AND FERTILIZER RECOMMENDATION USING MACHINE LEARNING ALGORITHMS

Authors-Devdatta A. Bondre, Mr. Santosh Mahagaonkar

The aim of proposed system is to help farmers to cultivate crop for better yield. The crops selected in this paper are based on important crops from selected location. The selected crops are Rice, Jowar, Wheat, Soyabean, and Sunflower, Cotton, Sugarcane, Tobacco, Onion, Dry Chili etc. The dataset of crop yield is collected from last 5 years from different sources. There have classified their process into 3 steps:

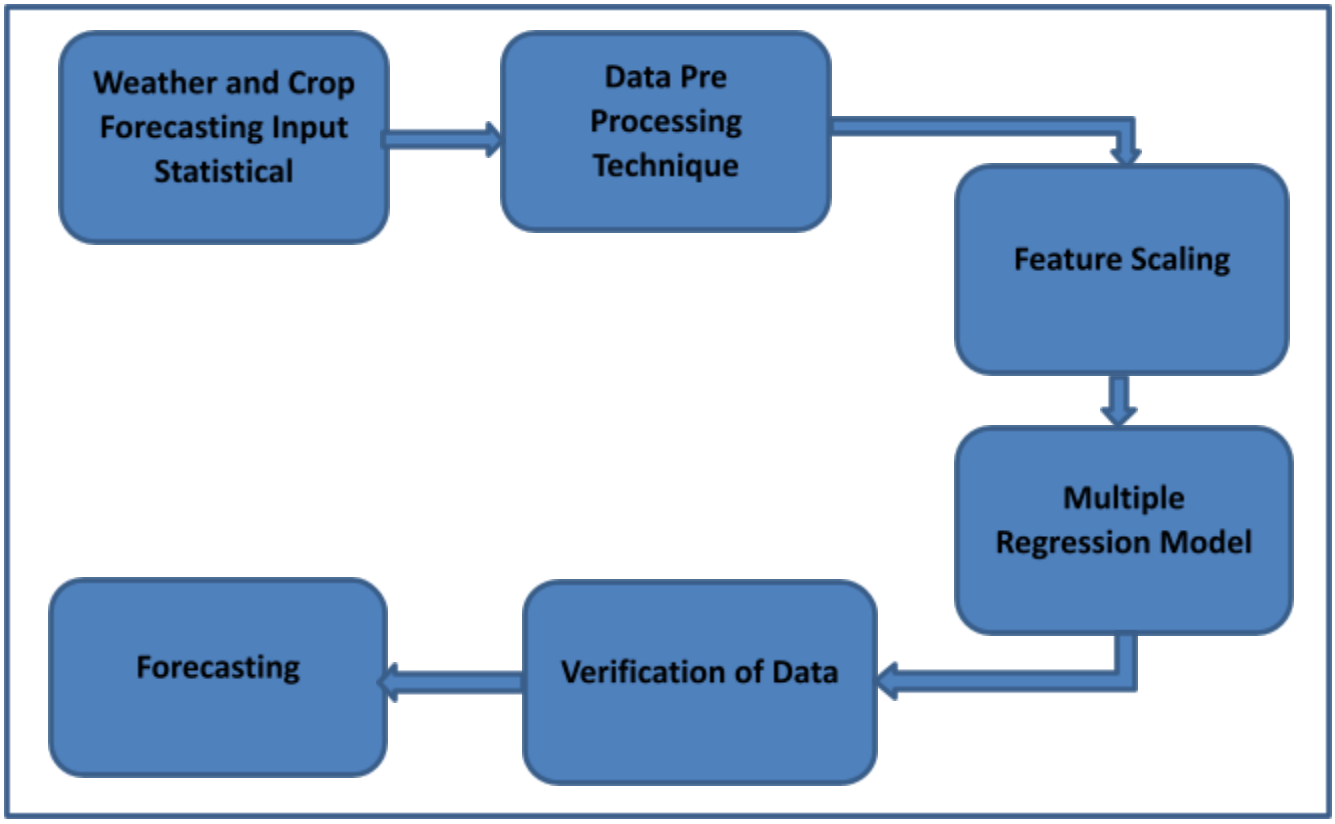
1) Soil Classification: Soil classification was done using soil nutrients data. Two Machine learning algorithms used for soil classification are Random Forest and Support Vector Machine. The two

algorithms classified, and displayed confusion matrix, Precision, Recall, f1-score and average values, and at the end accuracy in percentage as output.
2) Crop Yield Prediction: Crop Yield Prediction is done using crop yield data, nutrients and location data. These inputs are passed to Random Forest and Support Vector Machine algorithms. These algorithms will predict crop based on present input.
3) Fertilizer Recommendation: Fertilizer Recommendation can be done using fertilizer data, crop and location data. In this part suitable crops and required fertilizer for each crop is recommended.
· Third Party applications are used to display Weather information, Temperature information as well as Humidity, Atmospheric Pressure and overall description. Key takeaway from this paper was Random Forest is good with accuracy 86.35% compare to Support Vector Machine. For crop yield prediction Support Vector Machine is good with accuracy 99.47% compare to Random Forest algorithm. The work can be extended further to add following functionality. Mobile application can be built to help farmers by uploading image of farms. Crop diseases detection using image processing in which user get pesticides based on disease images. Implement Smart Irrigation System for farms to get higher yield

Methodology:

To find various solutions in machine learning the methodology also should be very simple. It contains six steps in which the first step concentrates on input statistical data which is weather and forecasting crop yield dataset. The second phase, Data pre-processing techniques involves transforming the raw data into understandable format. After pre-processing techniques, the next phase is dimensional reduction which is used to reduce the number of random variables under consideration by obtaining a set of principal variables and data scaling done. The entire data set then undergo KNN regression algorithm in which verification of data and forecasting should be done to achieve good results. The fertilizer recommendation system is developed using the KNN Classifier and is done on the basis of the required yield.

Methodology Flow Chart



Tools and Technologies:

The model is developed using Python, it's the best fit for machine learning and AI-based projects because of its simplicity and consistency, access to great libraries and frameworks for AI and machine learning (**ML**), flexibility, platform independence, and a wide community.

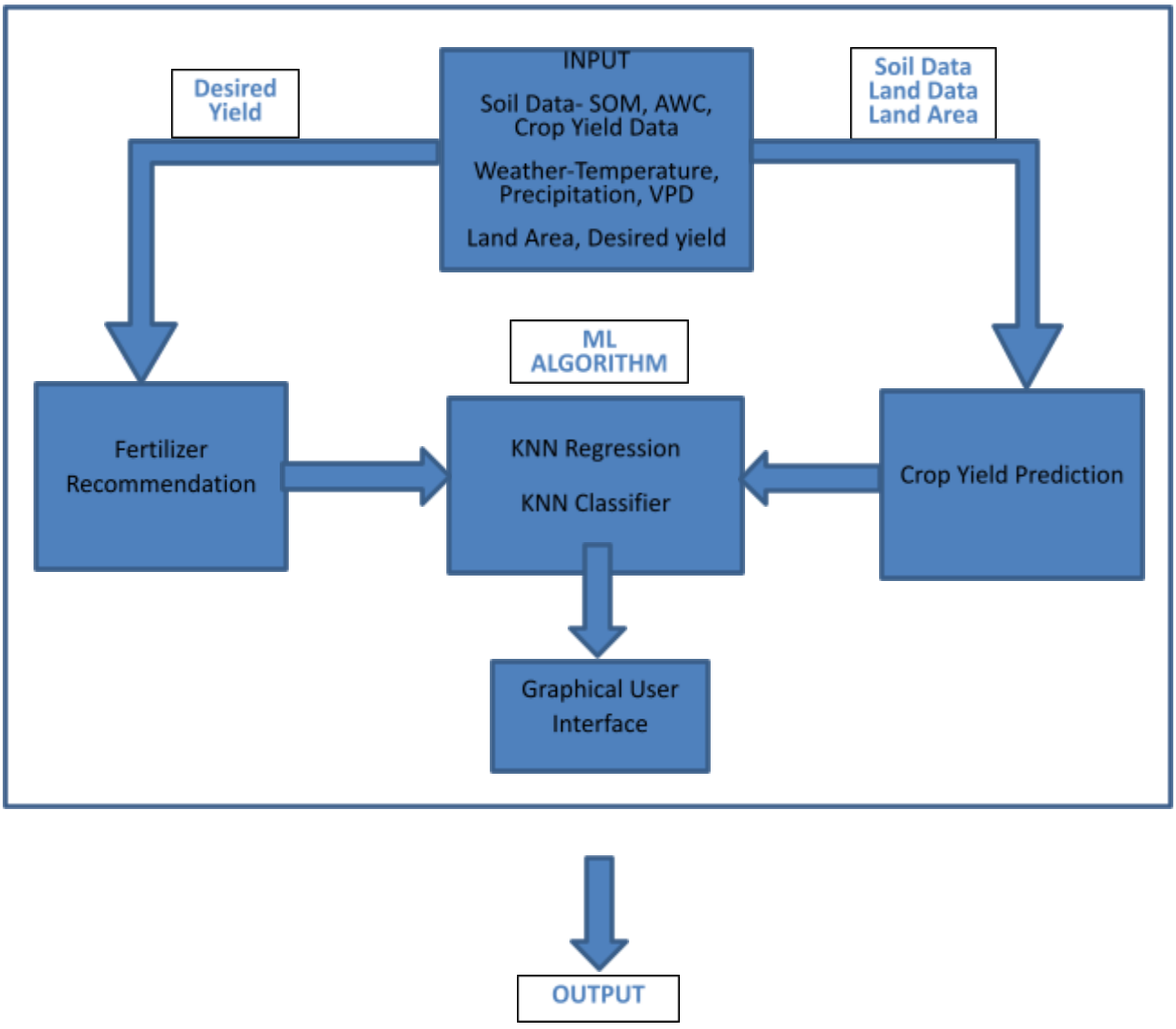
The libraries we used inn python were:

- NumPy
- Pandas
- Matplotlib
- Seaborn
- SciPy
- Scikit-learn
- Streamlit

The machine learning models used :

- K- Neighbors Regression.
- K- Neighbors Classifier.

Architectural Diagram of the Project



Yield Prediction Model Implementation:

Exploratory Data Analysis:

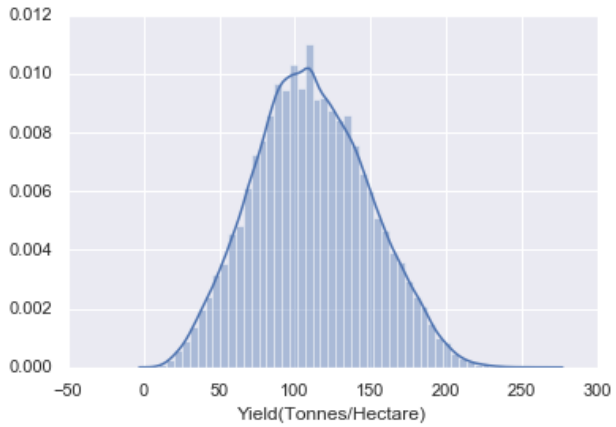
- Data Set: The screenshot shows 5 columns of the data set used and all the features used for yield prediction.

```
In [52]: dat.head()
```

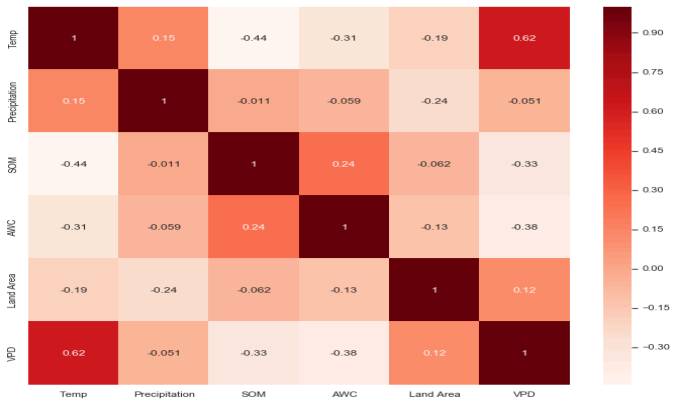
Out[52]:

	Temp	Precipitation	SOM	AWC	Land Area	VPD	Yield(Tonnes/Hectare)
0	20.094990	58.196000	1.246915	0.148338	436036.480	10.490231	30.0
1	20.089992	66.334063	1.464472	0.145533	424346.048	10.957824	30.9
2	20.460485	77.305455	1.477992	0.142567	623394.944	10.875867	49.0
3	19.560460	54.228760	1.386158	0.155162	571869.760	10.147539	55.2
4	20.237434	73.198760	1.345140	0.152290	583355.968	10.643400	53.2

- It shows the distribution of Yield (Tones/Hectare) against the density distribution

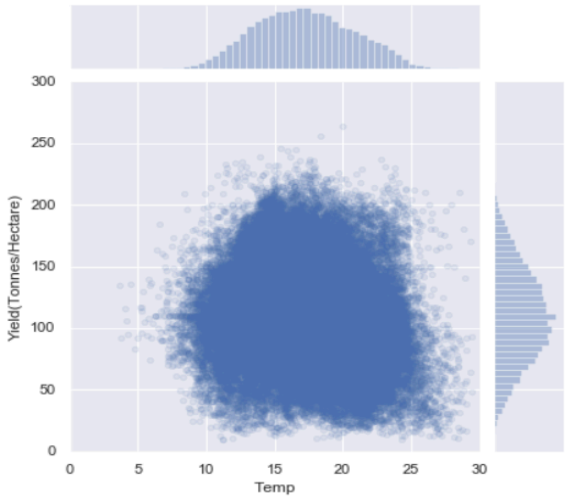


- It shows the correlation between features and is a heat map for better visualization.



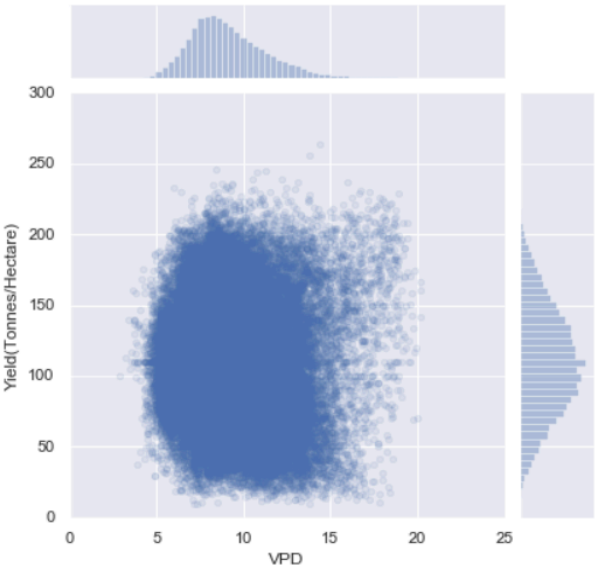
- The correlation scatter plot between temperature and Yield is shown to verify its dependence on yield.

```
In [55]: sns.jointplot(x='Temp',y='Yield(Tonnes/Hectare)',data=dat,kind='scatter',alpha=0.2)
         cor,_=pearsonr(dat['Yield(Tonnes/Hectare)'],dat['Temp'])
         cor

Out[55]: -0.17486337985063663
```



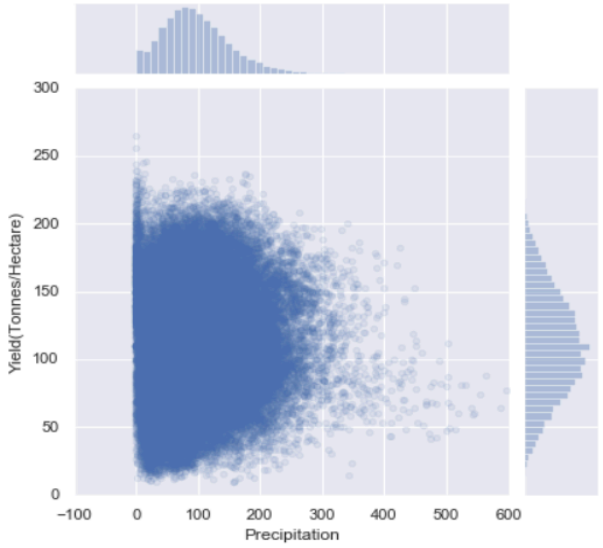
- The correlation scatter plot between Vapor pressure Deficit and Yield is shown to verify its dependence on yield.

```
In [56]: sns.jointplot(x='VPD',y='Yield(Tonnes/Hectare)',data=dat,kind='scatter',alpha=0.1)
         cor,_=pearsonr(dat['Yield(Tonnes/Hectare)'],dat['VPD'])
         cor

Out[56]: -0.068811515511878
```



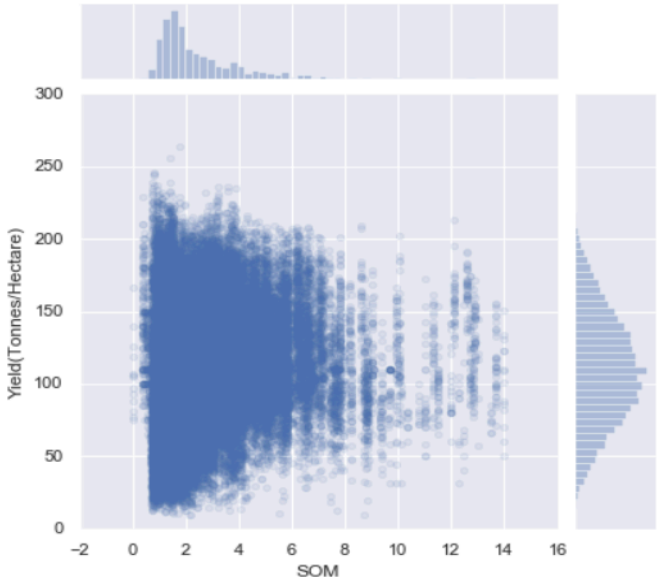
- The correlation scatter plot between Precipitation and Yield is shown to verify its dependence on yield.

```
In [58]: sns.jointplot(x='Precipitation',y='Yield(Tonnes/Hectare)',data=dat,kind='scatter',alpha =0.2)
         cor,_=pearsonr(dat['Yield(Tonnes/Hectare)'],dat['Precipitation'])
         cor

Out[58]: 0.0662030416384149
```



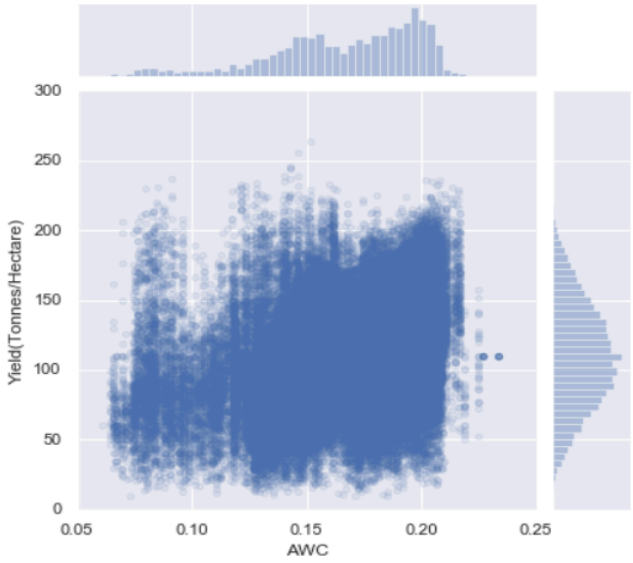
- The correlation scatter plot between Soil Organic matter and Yield is shown to verify its dependence on yield.

```
In [59]: sns.jointplot(x='SOM',y='Yield(Tonnes/Hectare)',data=dat,kind='scatter',alpha=0.2)
         cor,_=pearsonr(dat['Yield(Tonnes/Hectare)'],dat['SOM'])
         cor

Out[59]: 0.1358322712483059
```



- The correlation scatter plot between Available Water Capacity and Yield is shown to verify its dependence on yield.

```
In [60]: sns.jointplot(x='AWC',y='Yield(Tonnes/Hectare)',data=dat,kind='scatter',alpha=0.2)
         cor,_=pearsonr(dat['Yield(Tonnes/Hectare)'],dat['AWC'])
         cor
Out[60]: 0.3093409209722697
```



Algorithm:

- The values of the features ranged over a very big scale so data scaling is done on the features in order to bring the values of the features within a specific scale.

```
In [115]: from sklearn.preprocessing import StandardScaler
          # Initialise the Scaler
          scaler = StandardScaler()

          # To scale data
          scaler.fit(dat.drop(['Yield(Tonnes/Hectare)'],axis=1))
          scaled_features = scaler.transform(dat.drop(['Yield(Tonnes/Hectare)'],axis=1))
```

```
In [116]: df_feat = pd.DataFrame(scaled_features,columns=dat.columns[:-1])
          df_feat.head()
```

Out[116]:

	Temp	Precipitation	SOM	AWC	Land Area	VPD
0	0.798587	-0.708027	-0.739753	-0.553980	-0.111759	0.576647
1	0.797212	-0.567322	-0.613464	-0.640780	-0.135758	0.782791
2	0.899114	-0.377628	-0.605616	-0.732602	0.272867	0.746659
3	0.651567	-0.776620	-0.658925	-0.342747	0.167091	0.425567
4	0.837765	-0.448632	-0.682735	-0.431648	0.190671	0.644173

- Now the data is split into two parts as 70% for training and 30% to testing the algorithm.
  The image shows the code for the KNN regression algorithm being run multiple times for values of k ranging from 1-40 in order to find out the best fitting value of k.

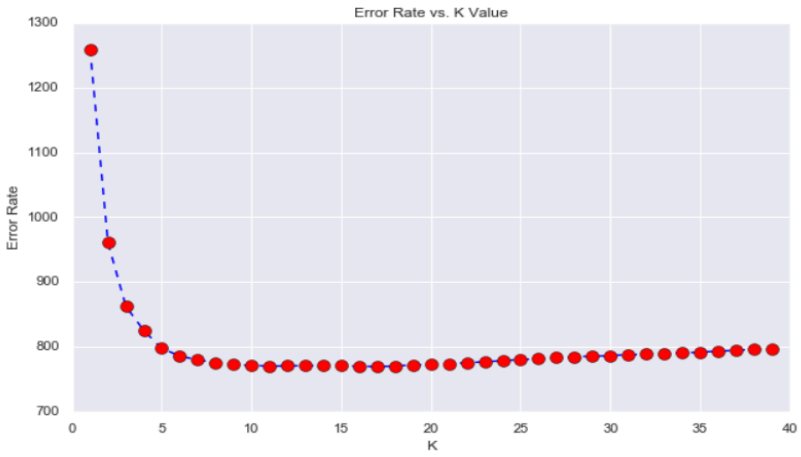
```
In [118]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
In [127]: error_rate = []
          from sklearn import metrics


          # Will take some time
          for i in range(1,40):

              knn = KNeighborsRegressor(n_neighbors=i,metric='euclidean')
              knn.fit(X_train,y_train)
              pred_i = knn.predict(X_test)
              error_rate.append(metrics.mean_squared_error(pred_i,y_test))
```

- The plot of the Error value for different values of k is drawn and the value of k corresponding to minimum error is selected.

```
In [128]: plt.figure(figsize=(10,6))
          plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
                   markerfacecolor='red', markersize=10)
          plt.title('Error Rate vs. K Value')
          plt.xlabel('K')
          plt.ylabel('Error Rate')
```

```
Out[128]: Text(0, 0.5, 'Error Rate')
```

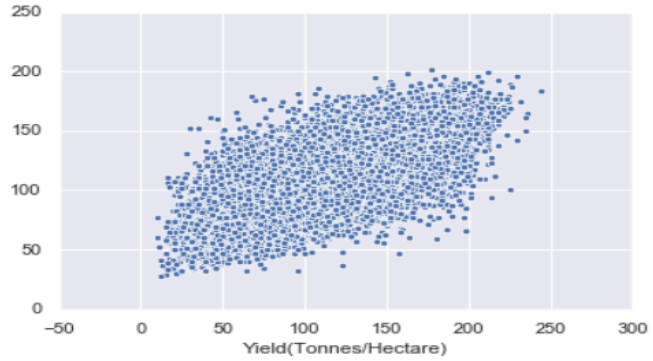


- Now the KNN Regression Algorithm in carried out for the best value of k i.e. 17 and the correlation scatter plot is made between the predicted value and the test value.

In [133]:
```
knn = KNeighborsRegressor(n_neighbors=17,metric='euclidean')
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
```

In [132]:
```
sns.scatterplot(x=y_test,y=y_pred)
cor,_=pearsonr(y_test,y_pred)
print("Correlation b/w Predicted value and test value = ",cor)
```

```
Correlation b/w Predicted value and test value =  0.6833632618854906
```



Efficiency of the Algorithm using different metrics:

The efficiency of the algorithm is tested using different metrics like mean absolute error, mean squared error and root mean squared error. The accuracy of the model is found to be 80.41%.

Model Evaluation

In [140]:
```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
print("Coefficient of determination (R2 score) =", round(metrics.r2_score(y_test, y_pred), 2))
print("Accuracy(Based on MAE)= ", (((110.987716-metrics.mean_absolute_error(y_test, y_pred))/110.987716)*100),"%")
```

```
MAE: 21.741773075920364
MSE: 769.2335200182737
RMSE: 27.7350594017441
Coefficient of determination (R2 score) = 0.47
Accuracy(Based on MAE)=  80.41064916056084 %
```

Fertilizer Recommendation Model Implementation:

Exploratory Data Analysis:

- Data Set: We obtained the list of fertilizers and corresponding maize yield.The screenshot shows 3 columns of the data set used and all the features used for fertilizer recommendation.

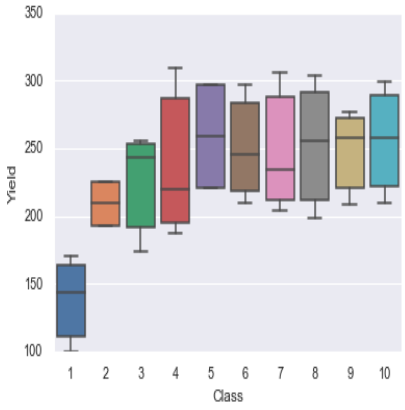
Class	Fertilizer(N-P-K)
1	0-0-0
2	44-15-17
3	46-15-25
4	69-15-25
5	69-30-40
6	80-15-40
7	80-30-0
8	80-30-25
9	80-30-40
10	92-30-40

	A	B	C
1	Fertilzers	Yield	Class
2	0–0–0	170	1
3	0–0–0	100	1
4	0–0–0	144	1
5	0–0–0	170	1
6	0–0–0	100	1
7	0–0–0	144	1
8	44–15–17.	225	2
9	44–15–17.	193	2
10	44–15–17.	225	2
11	44–15–17.	193	2
12	46–15–25	256	3
13	46–15–25	174	3
14	46–15–25	243	3
15	46–15–25	256	3
16	46–15–25	174	3
17	46–15–25	243	3
18	69–15–25	309	4
19	69–15–25	187	4
20	69–15–25	220	4

- The figure below shows the box plots and count plots of the distribution of different class fertilizer in the dataset.

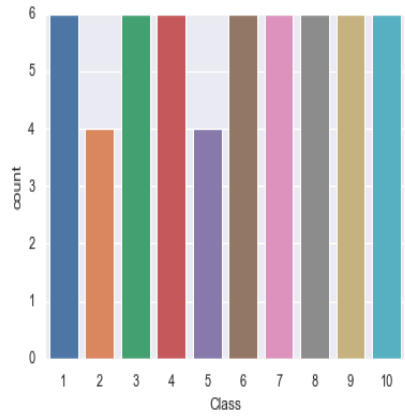
```
In [160]: sns.boxplot(data=dfs,y='Yield',x='Class')
```

```
Out[160]: <matplotlib.axes._subplots.AxesSubplot at 0x2f1f03ec208>
```




```
In [164]: sns.countplot(data=dfs,x='Class')
```

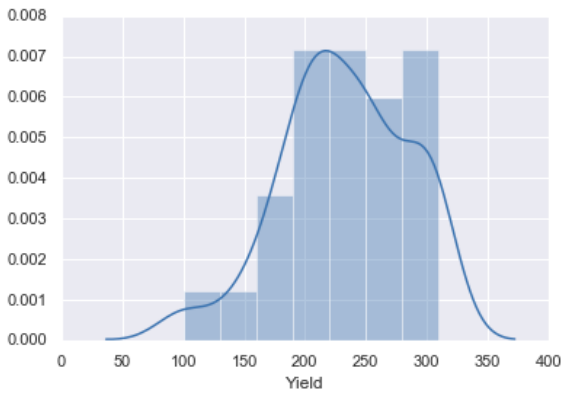
```
Out[164]: <matplotlib.axes._subplots.AxesSubplot at 0x2f1ebfda7b8>
```



- The figure below shows the yield distribution according to different class of fertilizers.

```
In [170]: sns.distplot(dfs['Yield'])
```

```
Out[170]: <matplotlib.axes._subplots.AxesSubplot at 0x2f1f569c7b8>
```



Algorithm:

- The data is split into two parts as 70% for training and 30% to testing the KNN Classifier Algorithm.

```
In [190]: X_train, X_test, y_train, y_test = train_test_split(dfs.drop('Class',axis=1),
                                                    dfs['Class'], test_size=0.15,
                                                    random_state=101)
          from sklearn.neighbors import KNeighborsClassifier
```

```
In [191]: error_rate = []
          from sklearn import metrics


          # Will take some time
          for i in range(1,40):

              knn = KNeighborsClassifier(n_neighbors=i)
              knn.fit(X_train,y_train)
              pred_i = knn.predict(X_test)
              error_rate.append(np.mean(pred_i != y_test))
```

- The image shows the code for the KNN regression algorithm being run multiple times for values of k ranging from 1-40 in order to find out the best fitting value of

k. Looking at the graph we concluded that at k=1 the error rate for our model is lowest.

```
In [192]: plt.figure(figsize=(10,6))
          plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
                   markerfacecolor='red', markersize=10)
          plt.title('Error Rate vs. K Value')
          plt.xlabel('K')
          plt.ylabel('Error Rate')

Out[192]: Text(0, 0.5, 'Error Rate')
```

Efficiency of the Algorithm using different metrics:

In [195]:

```
# FIRST A QUICK COMPARISON TO OUR ORIGINAL K=1
knn = KNeighborsClassifier(n_neighbors=1)

knn.fit(X_train,y_train)
pred = knn.predict(X_test)

print('WITH K=1')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))
```

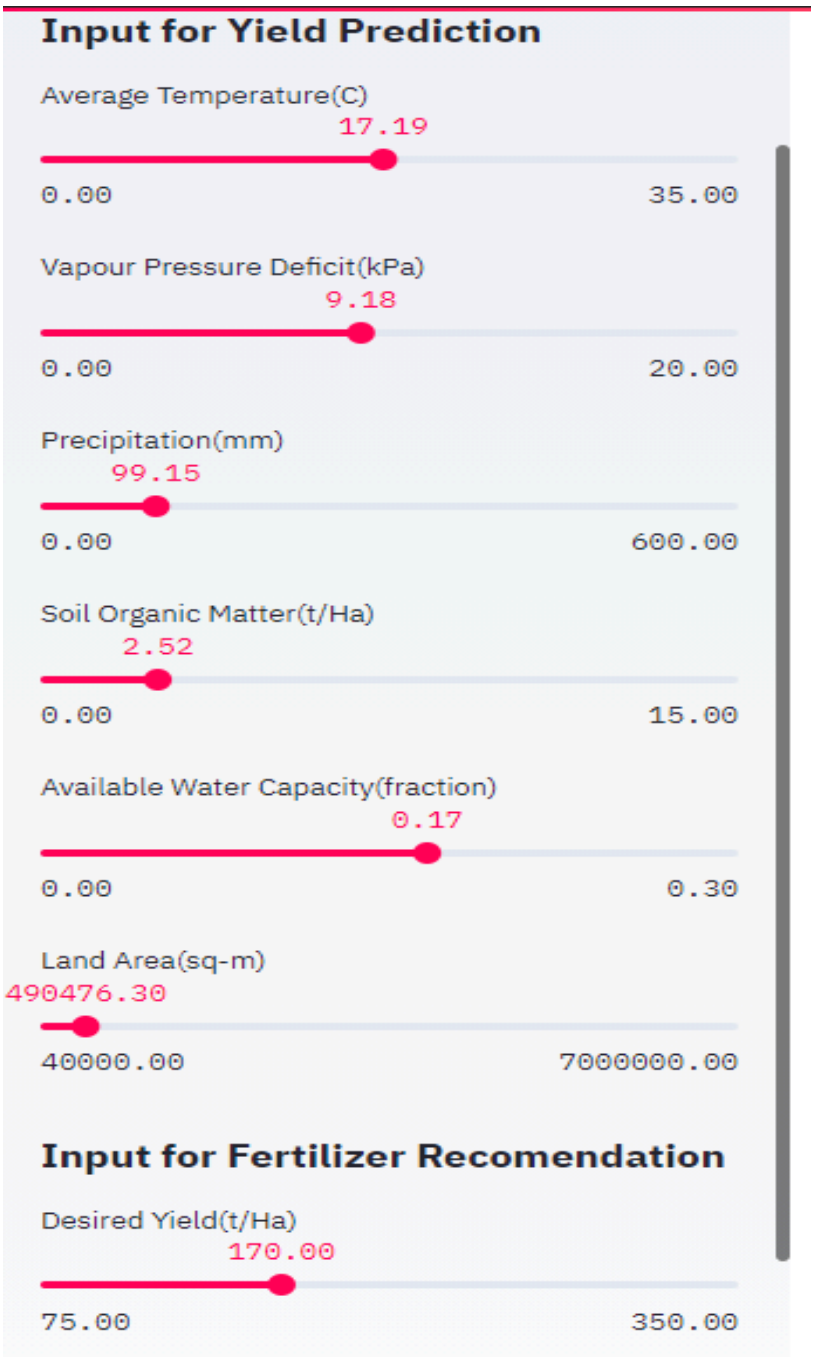
```
WITH K=1


[[1 0 0 0 0 0 0]
 [0 2 0 0 0 0 0]
 [0 0 1 0 0 0 0]
 [0 0 1 0 0 0 0]
 [0 0 0 0 2 0 0]
 [0 0 0 0 0 0 1]
 [0 0 0 0 0 0 1]]

              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1
           4       1.00      1.00      1.00         2
           5       0.50      1.00      0.67         1
           6       0.00      0.00      0.00         1
           8       1.00      1.00      1.00         2
           9       0.00      0.00      0.00         1
          10       0.50      1.00      0.67         1

    accuracy                           0.78         9
   macro avg       0.57      0.71      0.62         9
weighted avg       0.67      0.78      0.70         9
```

At k=1 testing the model the accuracy obtained was 78% on the basis of f1-score.

GUI of the project:

The GUI is developed using streamlit library of python. Streamlit's open-source app framework is the easiest way for data scientists and machine learning engineers to create beautiful, performant apps in only a few hours! All in pure Python.

INPUT-



OUTPUT-

Crop Yield Prediction and Fertilizer Recommendation App

This app predicts the **Crop Yield** and recommends **Fertilizer** to be used on the basis of Weather, Soil Parameters and Desired Yield

User Input Parameters

Yield Prediction

	Temp	VPD	Precipitation	SOM	AWC	Land Area
0	17.1915	9.1822	99.1465	2.5213	0.1662	490,476.3000

Fertilizer Recommendation

	Desired Yield(t/Ha)
0	170

Result

Predicted Yield(t/Ha)

	0
0	122.3818

Accuracy-80.40%

Recommended Fertilizer(N-P-K)

0-0-0

Accuracy-78.00%

RESULTS AND DISCUSSION:

There are various frameworks that use different systems to control information, to determine bits of knowledge and help hesitation making for ranchers. In any case, the significant concern is that they center either around one harvest expectation or gauge anybody parameter like either yield or cost. The report is a detailed account on the idea and working to implement the concept of intelligent analytics in the agriculture domain. The research work provides the information about how to apply data analytics on Maize crop datasets. From research and government data, a consolidated dataset was processed before applying machine learning concepts to it. These datasets include several parameters which are helpful to know the condition of maize crop and process them. This system has the capability to perform both the classification as well as regression. A KNN regression model was used to pass the dataset. We just need to pass the datasets through this system but the dataset should be in consistent form. This research work can be enhanced to the next level. We can build a recommender system of agriculture production and distribution for farmers. By which farmers can make decisions in which season which crop should sow so that they can get more benefit. This system works for structured datasets. In the future we can implement a data independent system also. It means format of data whatever, our system should work with the same efficiency.

References:

[1]. J. Ramirez-Villegas and A. Challinor, “Assessing relevant climate data for agricultural applications,” Agricultural Forest Meteorology, 2012, vol. 161(3), pp. 26–45. © 2018, IRJET | Impact Factor value: 7.211 | ISO 9001:2008 Certified Journal | Page 3159

[2]. C. O. Stockle., S. A. Martin and G. S. Campbell, “CropSyst, a cropping systems simulation model: water/nitrogen budgets and crop yield,” Agricultural Systems, 1994, vol. 46(3), pp. 335-359.

[3]. X. K. Chen and C.H. Yang, “Characteristic of agricultural complex giant system and national grain output prediction,” System Engineering Theory and Practice, 2002, vol. 6(6), pp. 120-125.

[4]. J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Operating Systems Design & Implementation, 1989, vol. 51(1), pp. 147–152.

[5]. J. Durbin, “Introduction to state space time series analysis,” State Space & Unobserved Component Models, 2004, pp. 3-25.

[6]. Wu X, Kumar V, Quilan JR, Ghosh J, Yang Q, Motoda H, McLanchlan GJ, Ng A, Liu B, Yu PS, Zhou Z-H, Steinbach M, Hand DJ, Steinberg D, Top 10 algorithms in data mining. KnowlInfSyst14: 1-37, 2008.

[7]. Abdullah, A., Brobst, S., M.Umer M. 2004. "The case for an agri data ware house: Enabling analytical exploration of integrated agricultural data". Proc. of IASTED International Conference on Databases and Applications. Austria. Feb

[8]. Abdullah, A., Brobst, S, Pervaiz.I.,Umer M.,A.Nisar. 2004. "Learning dynamics of pesticide abuse through data mining". Proc. of Australian Workshop on Data Mining and Web Intelligence, New Zealand, January.

[9]. Abdullah, A., Bulbul.R., Tahir Mehmood. 2005. "Mapping nominal values to numbers by data mining spectral properties of leaves". Proc. of 3 rd. International Symposium on Intelligent Information Technology in Agriculture. Beijing, China. Oct, 2005

[10]. Georg Ruß, Rudolf Kruse, Martin Schneider, and Peter Wagner. Estimation of neural network parameters for wheat yield prediction. In Max Bramer, editor, Artificial Intelligence in Theory and Practice II, volume 276 of IFIP International Federation for Information Processing, pages 109–118. Springer, July 2008

[11]. Applying Naive Bayes Data Mining Technique for Classification of Agricultural Land Soils P.Bhargavi, Dr.S.Jyothi, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August 2009 117

[12]. A. Mucherino, A. Urtubia, Consistent Biclustering and Applications to Agriculture, IbaI Conference Proceedings, Proceedings of the Industrial Conference on Data Mining (ICDM10), Workshop “Data Mining in Agriculture” (DMA10), Berlin, Germany,105-113, 2010.

[13]. Tripathi S, Srinivas VV, Nanjundiah RS Downscaling of precipitation for climate change scenarios: a Support Vector Machine approach. J Hydrol ss330:621–640, 2006

[14]. Fagerlund S Bird species recognition using Support Vector Machines. EURASIP J Adv Signal Processing, Article ID 38637, p 8, 2007.

[15]. Yue Jin Hai, Song Kai, 2010. "IBLE Algorithm in agricultural disease diagnosis". In third International Conference on Intelligent Networks and Intelligent Systems held at Shenyang, Liaoning China during November 01- November 2003