International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 224-230 www.ijaem.net ISSN: 2395-5252

Crop Recommendation on the basis of Temperature and AQI Prediction

Arushi Dheer M.L.Sharma , Krishna Chandra Tripathi, Dr.Neelu Dheer
Information TechnologyInformation Technology Maharaja Agrasen Institute of Maharaja AgrasenInstitute of Technology, Delhi, India Technology, Delhi, India
Information Technology Maharaja Agrasen Institute of Technology A.N.D.C, Delhi, India
Chemistry dept Delhi University

Submitted: 10-06-2021 Revised: 23-06-2021 Accepted: 26-06-2021

ABSTRACT
The temperature and Air quality index plays a key role in the growth of a crop. Due to changing climate patterns and increasing global warming, temperature around the world is on the rise and during these rapid changes, India is witnessing climate anomalies that are affecting its agricultural growth. Extreme high temperatures reduce the net gain of crop yield while low temperatures also stunt reproductive growth. Similarly, the concentration of pollutants stunts the yield. In this context, this paper analyzes historical data of temperature and AQI for Delhi, India and processes this data through a Time series algorithm and collaborative filtering. Based on the results of the model, it recommends whether or not the temperature and AQI are suitable for crop production.

Keywords: Time series, Air Quality Index, Temperature, collaborative filtering, Crop recommendation

I. INTRODUCTION

Weather conditions play a key role in crop production and management system. Crop production is a function of weather parameters. Among these, Temperature and Air Quality Index most notably impact the yield of a crop. For a farmer to make an informed decision, the weather forecast is very important. Traditionally, farmers in India rely on their experience of specific area's weather patterns to grow a particular crop and most of the agricultural activities in India are still lacking modern technological advancements, given the fact that India is in grip of smog and extreme temperature variations. Major cities of India have witnessed extreme weather conditions such as smog and heatwaves. In this context, current agricultural yield can be made sustainable If farmers are fed with the information on whether or not to grow a crop in the upcoming season. To gain such insights, it is important to have information on two key weather parameters (temperature and AQI. It is in this context, I made use of a Machine learning algorithm (Time series) to accurately forecast the suitability of temperature and Air quality index for the production and management of crops. Depending on the output of the model, farmers can know whether the upcoming weather pattern (temperature and air quality index) is suitable or not for growing a particular crop. This paper uses the timer series data of Delhi, India for the Time series algorithm using the C# language.

II. LITERATURE REVIEW

To forecast weather has remained a daunting task for a researcher as it involves complex interactions of the parameters and these parameters are dynamic in nature and they are multi-dimensional. Weather variables like Humidity, AQI, Temperature continue to change their value from time to time. This variation over time (Timeseries) can be studied through a statistical technique to create a model that can forecast their next value accurately. The model would take data points in form of continuous data and has the following components associated with it: cyclic, seasonal, and trend. In past, traditional techniques were used to predict weather patterns such as numerical weather prediction as well as synoptic forecasting. The drawback of these old computer techniques is that these are not precise whereas statistical modeling of weather is more powerful in all respects compared to the conventional techniques. Regarding Timeseries, there are many models available such as ARIMA, SARIMA, VAR, VARMA, and SES. Artificial neural network ANN is commonly used by researchers to analyze the trend however it works

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 224-230 www.ijaem.net ISSN: 2395-5252

best for the dataset containing large volumes. Since this paper used small observations So ANN was ruled out in favor of ARIMA which is more robust. ARIMA is best suited to short-term forecasting. It efficiently establishes the relation between historical data and future trends.

III. METHODOLOGY

For time series, data points are spaced at regular intervals. It applies to any dataset whose values are changed over time (hours, days, months, or years). The dataset used in this paper contains a seasonality and trend component.

In the data preprocessing stage before the forecast happens, the average is taken on the dataset and then this average is pushed forward to the next data point.

Among moving average models, simple moving averages SMA and autoregressive integrated moving average models are common in use. SMA takes the sum of points in the given dataset and it then takes its average over a specified subset of data, giving the trend direction Whereas ARIMA uses a linear function to predict the past data. Since it relies on the past data, it subtracts the current values of data from the past data point (known as lag). It is a combination of autoregressive (AR) and moving averages (MA). The integration part combines both AR and MA into one model. If there is any noise in the model, it needs to be averaged out. If the purpose is to predict the value of the moving average, then the autocorrelation graph is plotted.

The coding part of the time series algorithm was done in a C sharp compiler in Microsoft visual studio which contains standard C# libraries for data manipulation. I first installed ML.NET.TimeSeries from the package‘s library of the visual studio.

I installed the lumen package to read CSV data from the Nuget package. This package is available in the visual studio. After adding the namespace, I wrote the following lines of code to read the CSV.

```
Var table=newDatatable();
using(varReadCsv=newCsvRreader(newStreamReader(openRead(@"C:/dataset.csv")))

{ table.load(ReadCsv)
 }
```

So, the table reads the data loaded from the CSV file. Any column of the table can be accessed by giving the index of the column.

Then I wrote a class to get and set the values in the table as shown below.

```
publicclass getsetvalues
{
publicstringInstance{get;set;}
publicstringTemperature{get;set;}
publicstring AQI{get;set;

}

```

There are two tests to check if the data is stationary or not. Through rolling statistics, I first defined the window for the dataset. Then, I calculated the rolling mean at a yearly level. Then I calculated the standard deviation using the std function. By analyzing the plot, it came out non-stationary.

I preprocessed the data set. I denoted temperature in centigrade, and AQI in micrograms per cubic meter. The dataset snapshot is given below:

ID	Temperature	AQI
50	42	256
49	42	463
48	39	187
47	34	271
46	38	266
45	37	342
44	38	214
43	39	240
42	42	177
41	41	302

40	40	151
39	39	107
38	37	64
37	35	60
36	34	82
35	36	46
34	35	61
33	35	90
32	35	136

In c sharp, I implemented the ARIMA(p,d,q) by constructing ArimaModel class where
P is the order of autoregression.
q is moving average
d is the degree of difference.

```
Var forecastmodel=newArimaModel(Data)
```

I used the following namespaces.

```
Using system ;// global namespace
UsingStatistics.TimeSeriesAnalysis;// namespace of statistics for sql server
```

Then I wrote the foreach value which iterates over the parameter values.

```
foreach(param in model.Parameters)
                    Console.WriteLine(
Name,Value,StandardError,Pvalue);
```

Here, the name denotes the name of the variable. Value is an estimated one. Pvalue is the probability of statistics.

The EstimateMean value was set to true because by default the mean is zero for the integrated model. Then, I headed towards the computational part of the code using the compute method.

```
Forecastmodel.compute();
```

The model is fit into the compute method.
After that, I created another class named TSeries to manipulate time-series data.

```
classTSeries
{var tempAqi = timeSeries.Resample(Recurrence.Monthly, aggregators);

}
```

Then, I used the forecast method to predict temperature and AQI suitability on demand. The code snippet is shown below.

```
var model= pipeline.Fit(data)
var forecastmodel= model.createTimeSeriesEngine(context);
var forecasts= forecastingEngine.Predict()

foreach(var f in forecasts.forecast)
{ console.WriteLine(forecast);
}
```

In the SQL database, I have the stored table which contains the suitable ranges of optimum ranges of temperature as well as the AQI for the given crop. Under these ranges, the crop gain will be maximum. Here is the snippet of this table.

Id	Crop TYPE	Max suitable temperature	Maximum suitable AQI	Max suitable temperature	Maximum suitable AQI
1	Rice	38	256	20	240
2	wheat	35	20	20	15
3	Apple	30	30	20	80

Table: optimum suitable ranges of temperature and AQI

Regarding the recommender system, there are two methods commonly used: content-based approach and collaborative filtering. For the content-based approach, it needs an ample amount of data to predict behavior while Collaborative filtering only requires historical preferences. So, I made use of collaborative filtering (CF) which is a technique to make automatic predictions and is often used in recommender systems. It is either based on memory or model. Model-based filtering is used in data mining or machine learning applications. It is also called the nearest neighbor algorithm. It takes the outcome of prediction from the model and cross-refers the predicted data with historical data in the Microsoft SQL server management table and filters out the result. I made the database named CropYieldPrediction. In this, I created two tables named CropType and Dataset.

$$\mathrm{Cf} = \sum \mathrm{similaritiesdividedbynumberofrating}$$

The cosine similarity method is most famous to arrive at CF value. It gives weightage to the ratings based on prediction. In our case, ratings are based on temperature and AQI. The benefit of using the CF rating method is that ratings do not drastically change over time. It empowers the predictor system of time series.

For determining the trend direction, I calculated weighted moving averages. I multiplied each datapoint in the dataset with a fixed weighted factor. The highest weight is given to the most recent number.

$$\mathrm{weightedmovingaverage(temp)} = \mathrm{temp1} * \mathrm{n} + \mathrm{temp2} * (\mathrm{n} - 1) + \cdots . + \mathrm{temp(n)} \div [\mathrm{n} * (\mathrm{n} + 1)]/2$$

The same formula was repeated for the AQI as shown below.

$$\mathrm{weightedmovingaverage(AQI)} = \mathrm{AQI1} * \mathrm{n} + \mathrm{AQI2} * (\mathrm{n} - 1) + \cdots . + \mathrm{AQI(n)} \div [\mathrm{n} * (\mathrm{n} + 1)]/2$$

Then I tabulated the results which contain the following headers: Instance, value, forecast, error.

Instance	Value	Forecast	error
0	42	42	0
1	42	42	0
2	39	42	3
3	34	41	7
4	38	39.25	2.25

Table 1 Simple moving averages (Temperature)

Instance	Value	Forecast	error
0	42	42	0
1	42	42	0
2	39	0	39
3	34	39.6	5.60

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 224-230 www.ijaem.net ISSN: 2395-5252

4	38	35.15	-2.85
5			

Table 2 Weighted moving averages (Temperature)

Instance	Value	Forecast	error
0	256	256	0
1	463	256	-207
2	187	359.6	172.5
3	271	302	31
4	266	294.25	28.25
5			

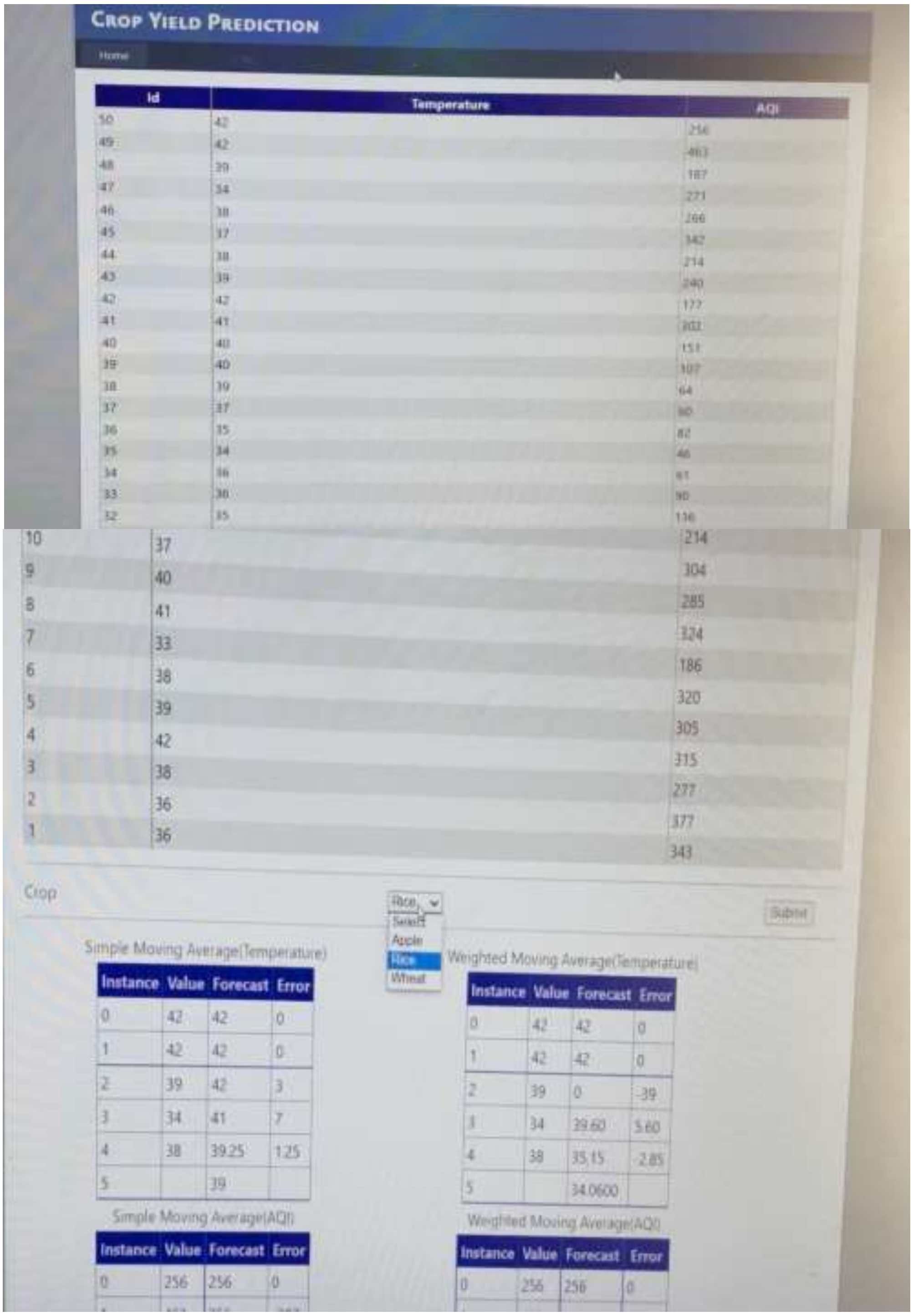
Table 3 Simple moving averages (AQI)

Instanc e	Value	Forecast	error
0	256	256	0
1	463	256	-207
2	187	0	-187
3	271	231.85	-38.15
4	266	268	2
5			

Table 4 weighted moving averages (AQI)

For both AQI and temperature, the above tables show forecasted values against the given instance. The error column is calculated by taking the difference between the forecast and the value column.

For purpose of illustrating the forecasted values and trends, the live web localhost was used. For this, I made use of asp.net web pages in Microsoft visual studio. On the aspx page (monitoring.aspx), I designed the front end of it using a form structure. It contained an interactive dropdown having crop names against which prediction was to be made. The crop names I fed into this dropdown were: apple, wheat, and rice.



After selecting the particular crop, the data (to be forecasted) is fed via submit button at the bottom right corner. Then, time-series calculations were performed and the next loaded screen popup gives the results of the prediction as shown below.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 7 July 2021, pp: 224-230 www.ijaem.net ISSN: 2395-5252

When predicted values fall into the suitable range for a crop, the results show that the given temperature value is suitable otherwise it shows the predicted value for a crop is not suitable.

IV. RESULTS

The objective was to recommend crops by estimating the trend over historical weather data. According to test results, temperature and AQI are best predicted by the time series algorithm. This work worked on the logic to determine the suitability of temperature and AQI for crops in Delhi using the range of continuous datasets rather than using discrete data. However, the error in output of the model does not pose any major problem, given the actual and estimate values remained in close vicinity. This model can satisfactorily determine the suitability of ranges of temperatures and quality index for the crop. The accuracy made by the model is satisfactory and fits into the desired goal. Further, by including more data points, the accuracy of this forecasted trend can be made better.

V. CONCLUSION

In this paper, a model of machine learning using the time series algorithm was employed to suggest weather parameters for the feasibility of the crop in Delhi, India. It kept the focus on two parameters: temperature and AQI. Other weather variables are equally capable of influencing the accuracy. By inputting the other weather parameters in future modeling, this study can be made more accurate to predict.

REFERENCES

(2021). Introduction to time series analysis. Algorithmia Blog, https://algorithmia.com/blog/introduction-to-time-series.
(2019). A simple introduction to collaborative filtering. built in , https://builtin.com/data-science/collaborative-filtering-recommender-system.
statistics for SQL server, https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/provider-statistics-for-sql-server.