Improvising the Nervous Market System: Agricultural Market Prices for the Farmers

Rudviq Bhavsar, Omkar Masal, Mohit Gadhia, Ritesh Patil, Prof. Prajwali Korde

B.E Student, Dept. of Information Technology, International Institute of Information Technology(I^{2}IT), Pune, India

B.E Student, Dept. of Information Technology, International Institute of Information Technology(I^{2}IT), Pune, India

B.E Student, Dept. of Information Technology, International Institute of Information Technology(I^{2}IT), Pune, India

B.E Student, Dept. of Information Technology, International Institute of Information Technology(I^{2}IT), Pune, India

Assistant Professor, Dept of Information Technology, International Institute of Information Technology(I 2 IT), Pune, India

ABSTRACT:-The majority of India relies on agriculture for livelihood, but sometimes farmers fail to produce the appropriate amount of crops for supply. Being unaware of the market condition, the farmer usually sells middlemen the crops as soon as it gets off the field. Middlemen take advantage of such a situation and buy crops for a very cheap amount resulting in loss of the farmer. If the same crops were sold to the customers at a higher rate when the demand's at the peak, the farmer would have earned profit. In this paper, we are attempting to predict the demand and price of the crop at a certain period by analyzing patterns from the past data so that farmers won't face huge losses. Our system uses 10-year-old previous data to predict demand for a further 3 months.

KEYWORDS: Android Application, FBprophet.

I. INTRODUCTION

Agriculture is one of the major sectors of the Indian economy and it is present for thousands of years. India largely depends on the agriculture sector. In recent years India has been observing an economical growth. This increase in the economy leads to an increase in incomes, indirectly resulting in an increase of demands for food commodities. It's the situation that every country faces due to growth in the economy[1][2][3].

Most of the farmers are dependent on agriculture as an occupation and the only option to fulfill their basic needs. Farmers cultivate and supply the crops to the venders. Being unaware of the market conditions, farmers trade with the vendors at a non-negotiable price which leads the farmer to lose. The vendor, on the other hand, store those crops until there is an increase in demand. When the demands increase, the price of the crop increases ultimately vendor earns a huge profit. Despite such efforts, the farmer faces loss.

The reason for the suffering of farmers is that they are unable to predict the demand and price of the crop that is going to hit the market. It is very necessary to introduce them with a medium so that they can easily predict the price of the crop for a particular day. To solve problemsof the farmers we have come up with an efficient way that can predict the price of the crop on any particular day. Fbprophet is an ultimate tool that can be used to suggest a solution for the farmer's benefit.

Forecasting is a method that uses historical data as input to make informed estimates that are predicted in determining future forecasting. It is an important and standard function of data science in organizations today. Having prior knowledge of any event can greatly help a company in the development of its goals, policies, and plans. However, producing high-quality and reliable forecasts presents its own challenges. Forecasting is a complex thing for humans and machines. It also requires highly experienced time series analysts which in fact is rare.

A prophet is a tool developed to address these issues and provides an effective way to guess at "scale". It aims to revolutionize the standard features of a business time series by providing simple and convenient methods. The prophet empowers critics in various domains to make predictions beyond what they can do manually.

The prophet is a process of estimating time series data based on a model of addition where nonlinear trends correspond to seasonal, weekly, and holiday results. It also works best with a time series with strong seasonal effects and a number of historical moments.

II. RELATED WORK

A literature survey was carried out to know about the actual reason for the increasing prices of the crops. The survey revealed that the amount of supply does not actually meet the demand in the market.

HuyVuong Nguyen et al. developed a time finite sensible system for value estimation supported statistic models. The theme performs the sequence of processes viz., Gathering transactions details from an internet site, pre-processing the collected unrefined info, associated mistreatment an ARIMA model. Root Mean Square Error (RMSE) and also the Mean Absolute Percentage Error (MAPE) used as performance metrics to assess the forecast exactitude strategy. The group action details of PriceMe web site is taken into account during this work. Moreover, the projected model is compared with the existing Moving Average (MA) model. the end result discloses that the forecast details of the projected model are time finite compared to MA model.[4]

Manpreet Kaur et al. learned about different data mining techniques viz., K-Means, K-Nearest Neighbour, Artificial Neural Networks and Support Vector Machines and its crop price prediction capability. Estimating crop prices using available data is of great need. Based on the research the author has introduced a model that helps farmers make the best decision to estimate crop prices accurately[5].

S. Rajeswari et. al. have studied and suggested the idea of pricing forecast based on big data and data mining using HADT Algorithm i.e. Hybrid Association Rule-based Tree Decision Algorithm. They have forecasted the price of the crop using the data of the Virudhunagar District, Tamilnadu. In particular, they used the R programming language to process agricultural data integrated with the Hadoop environment. Predictor specification was improved with the HADT model integrated with a few fizzy techniques to obtain the default module. [6]

III. METHODOLOGY

A. FBPROPHET

Prophet follows an analyst-in-the-loop approach to business forecasting at scale.

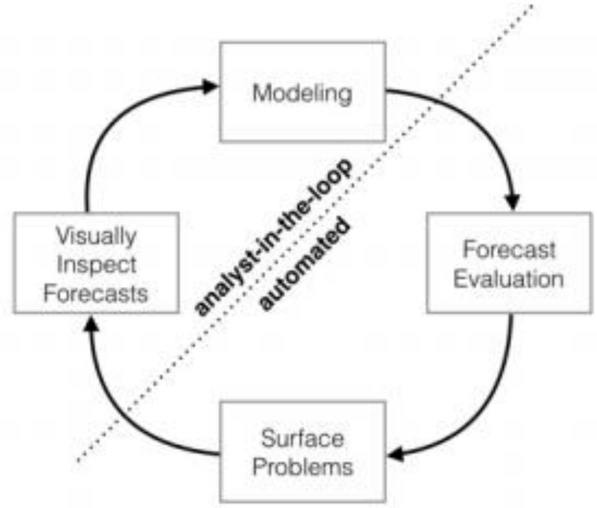


Fig 1.Schematic view of the analyst-in-the-loop approach to forecasting at scale

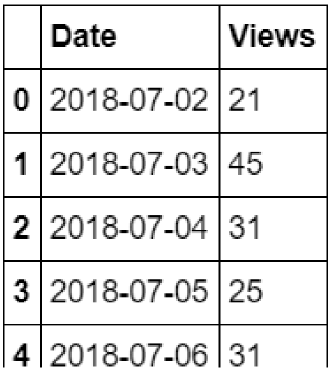
B.IMPORTING THE DATASET

The input to the Prophet is always a data column with two columns: ds and y. The column ds (datestamp) should be in the format expected by the Pandas, logically YYYY-MM-DD for the day or YYYY-MM-DD HH: MM: SS for the timestamp. Column y must be numeric and represent the desired ratio of prediction.

FBprophet was used to determine the views an article can have at a certain date. For this analysis, an excel file was used that contains the total ‘Views’ that articles have garnered.

```
importpandasaspd
importnumpyasnp
fromfbprophetimport Prophet
importmatplotlib.pyplotasplt
plt.style.use("fivethirtyeight")# for pretty graphsdf = pd.read_excel('medium_stats.xlsx')
df.head()
```

Table 1.Dataset



	Date	Views
0	2018-07-02	21
1	2018-07-03	45
2	2018-07-04	31
3	2018-07-05	25
4	2018-07-06	31

- **Analysing the datatypes**

```
df.dtypesDate     datetime64[ns]
Views            int64
dtype: object
```

- **Plotting to get insights**

```
plt.figure(figsize=(10,6))
plt.plot(df.set_index('Date'))
plt.legend(['Views'])
```

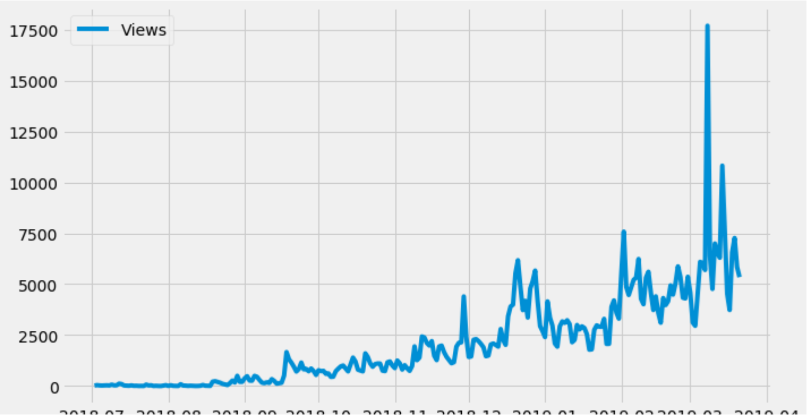


Fig 2. A plot of ‘Views’ with Time.

In Fig 2, the structure exhibits a high or low inclination but will have to carry out further processing to reach the concrete conclusion. Also, it looks like someone has released it except March’2019. In a look at the details, it was found that this appears around March 8th when the article was published: From ‘R vs Python’ to ‘R and Python’. There was a sudden outbreak that day with a peak of 17,688 views.

Converting Views column to its log value

```
df['Views'] = np.log(df['Views'])
plt.figure(figsize=(10,6))
plt.plot(df.set_index('Date'))
plt.legend(['Views'])
```

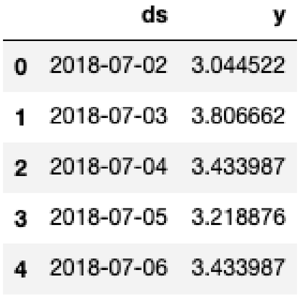
i) Making the predictions

To make the dataset 'Prophet' compatible.

Let's convert the data in the format desired by Prophet. We shall rename 'Date': 'ds' and 'Views': 'y'

```
df.columns = ['ds','y']
df.head()
```

Table 2. Data converted in format of 'ds' and 'y'



	ds	y
0	2018-07-02	3.044522
1	2018-07-03	3.806662
2	2018-07-04	3.433987
3	2018-07-05	3.218876
4	2018-07-06	3.433987

The Prophet follows the sklearn model API where the Prophet's class model is created and the appropriate and predictive methods are called. The model has been fitted with a new version of the Prophet and has been followed by calling its fit method and passing through the historical dataframe.

Seasonalities

By default the Prophet will be equal to certain weekly and yearly intervals if the time series is more than two cycles in length. And it will be the same as the daily season of the daily sub-series. You can add other seasonalities (monthly, quarterly, hourly) as needed.

```
m1 = Prophet(daily_seasonality=True)
m1.fit(df)
```

Forecasting

Prophet uses a linear model for its forecast by default. The Logistic model can also be used by passing it as an attribute.

```
#m = Prophet(growth='logistic')
#m.fit(df)
```

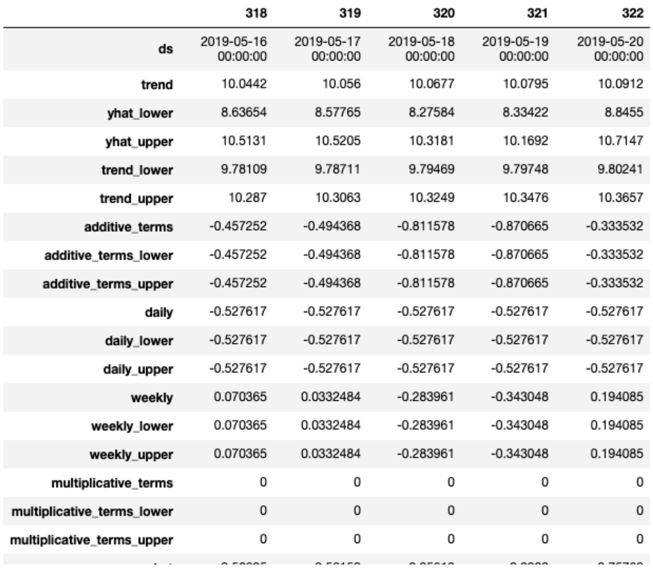
For forecasting, we need to tell the Prophet how far to predict in the future. In this case, we need to make the data set in the future forecast using make_future_dataframe. Here the prediction is considered for the next 60 days.

```
future1 = m1.make_future_dataframe(periods=60)
forecast1 = m1.predict(future1)
forecast1.tail().T
```

forecast is essentially a pandas dataframe and comprises of a lot of fields. The predict method will assign each row in future a predicted value which it names yhatand the range is defined as yhat_lower and yhat_upper . These ranges are considered as uncertainty levels.

To see the last 5 predicted values, the tail function is required . Here the Transpose function is brought to use to transpose the dataframe so that all the columns are visible. The Table 3 shows the predicted values from 16th May'2019 to 20th May'2019.

Table 3. Last 5 predicted values

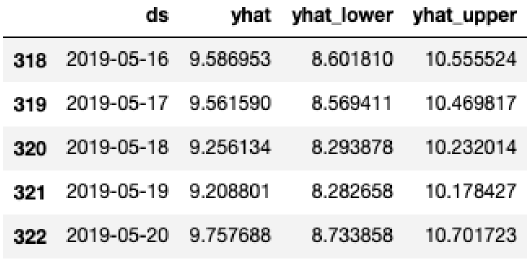


	318	319	320	321	322
ds	2019-05-16 00:00:00	2019-05-17 00:00:00	2019-05-18 00:00:00	2019-05-19 00:00:00	2019-05-20 00:00:00
trend	10.0442	10.056	10.0677	10.0795	10.0912
yhat_lower	8.63654	8.57765	8.27584	8.33422	8.8455
yhat_upper	10.5131	10.5205	10.3181	10.1692	10.7147
trend_lower	9.78109	9.78711	9.79469	9.79748	9.80241
trend_upper	10.287	10.3063	10.3249	10.3476	10.3657
additive_terms	-0.457252	-0.494368	-0.811578	-0.870665	-0.333532
additive_terms_lower	-0.457252	-0.494368	-0.811578	-0.870665	-0.333532
additive_terms_upper	-0.457252	-0.494368	-0.811578	-0.870665	-0.333532
daily	-0.527617	-0.527617	-0.527617	-0.527617	-0.527617
daily_lower	-0.527617	-0.527617	-0.527617	-0.527617	-0.527617
daily_upper	-0.527617	-0.527617	-0.527617	-0.527617	-0.527617
weekly	0.070365	0.0332484	-0.283961	-0.343048	0.194085
weekly_lower	0.070365	0.0332484	-0.283961	-0.343048	0.194085
weekly_upper	0.070365	0.0332484	-0.283961	-0.343048	0.194085
multiplicative_terms	0	0	0	0	0
multiplicative_terms_lower	0	0	0	0	0
multiplicative_terms_upper	0	0	0	0	0

However, we are only interested in theyhat, yhat_lower and yhat_uppervalues but only shown these values above to highlight the Prophet's ability to infer various values from the given data. Essentially, our desired dataframe will be as follows:

forecast1[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()

Table 4. using yhat, yhat_lower and yhat_upper

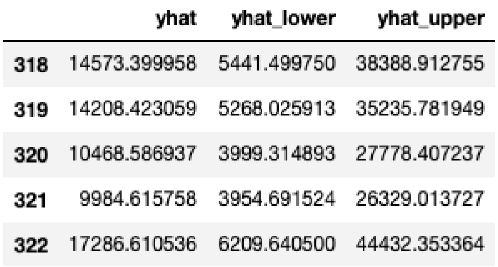


	ds	yhat	yhat_lower	yhat_upper
318	2019-05-16	9.586953	8.601810	10.555524
319	2019-05-17	9.561590	8.569411	10.469817
320	2019-05-18	9.256134	8.293878	10.232014
321	2019-05-19	9.208801	8.282658	10.178427
322	2019-05-20	9.757688	8.733858	10.701723

Since we took the log of Views, let's convert back to original values

np.exp(forecast1[['yhat', 'yhat_lower', 'yhat_upper']].tail())

Table 5. using yhat, yhat_lower and yhat_upper



	yhat	yhat_lower	yhat_upper
318	14573.399958	5441.499750	38388.912755
319	14208.423059	5268.025913	35235.781949
320	10468.586937	3999.314893	27778.407237
321	9984.615758	3954.691524	26329.013727
322	17286.610536	6209.640500	44432.353364

ii) Plotting the forecast

In Fig 3, the forecast is plotted by calling the Prophet.plot method and the forecast dataframe is passed. Plotting is done as visualization is more convenient than the numbers.

m1.plot(forecast1);

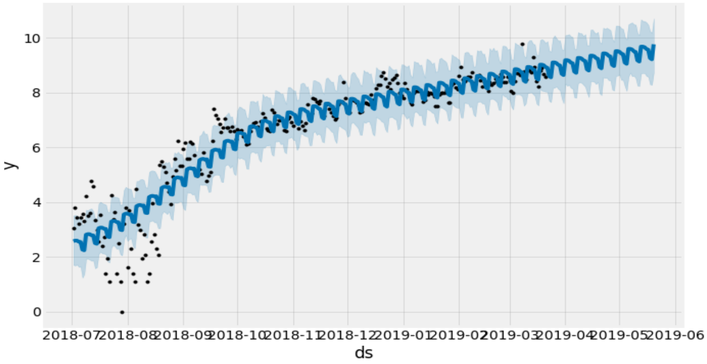


Fig 3. Plotting forecast for the next 60 days

Prophet.plot_components method can be used to see the forecast components which will show daily and weekly trends.This makes the picture clearer. If yearly trends is to be checked then yearly_seasonalitycan be enabled.

m1.plot_components(forecast1);

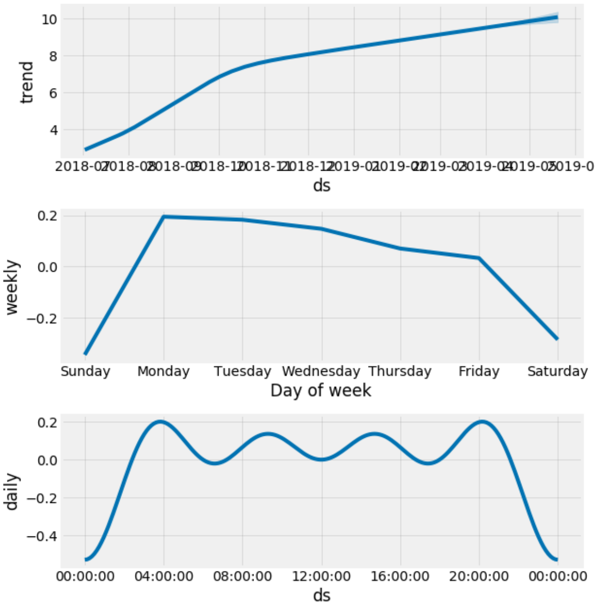


Fig 4. weekly and daily trends for the forecast

The trend shows an increase in views over a good period of time. Also, more views are collected from Monday to Friday than on weekends. The daily situation is somewhat constant.

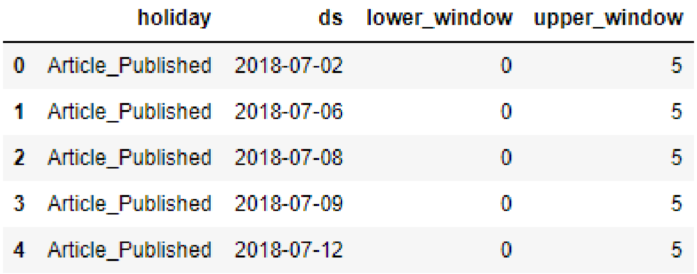
iii) Holiday effect

Holidays are times when days have the same impact each year. For example, if you want to simulate the number of subscribers in a city where people move in during the holidays, you can enter the dates of your model period using the holidays parameter.[8]

Holidays and events offer great shocks, something to be considered in many business time series and often do not follow the time pattern, so their effects are not exactly imitated by a smooth cycle. The Prophet offers the ability to incorporate the flexibility resulting from these holidays or special events into the forecasting model.

A sudden outbreak of the views was noticed on the day the article was published. This material lasts for about 4 to 5 days and then deteriorates. So the publication date can be considered as a holiday for the article. Also, aupper_window is defined of containing 5 days for which there are additional views.

“The Columns *lower_window* and *upper_window* extend the holiday out to *[lower_window, upper_window]* days around the date. For instance, if you wanted to include Christmas Eve in addition to Christmas you’d include *lower_window=-1,upper_window=0*. If you wanted to use Black Friday in addition to Thanksgiving, you’d include *lower_window=0,upper_window=1".*



	holiday	ds	lower_window	upper_window
0	Article_Published	2018-07-02	0	5
1	Article_Published	2018-07-06	0	5
2	Article_Published	2018-07-08	0	5
3	Article_Published	2018-07-09	0	5
4	Article_Published	2018-07-12	0	5

Once the creation of dataframe is done, holiday effects can be included in the forecast by passing them in with the holidaysargument and the forecasting process remains the same as above:

```
m2 = Prophet(holidays=articles,daily_seasonality=True).fit(df)
future2 = m2.make_future_dataframe(periods=90)
forecast2 = m2.predict(future2)
m2.plot(forecast2);
```

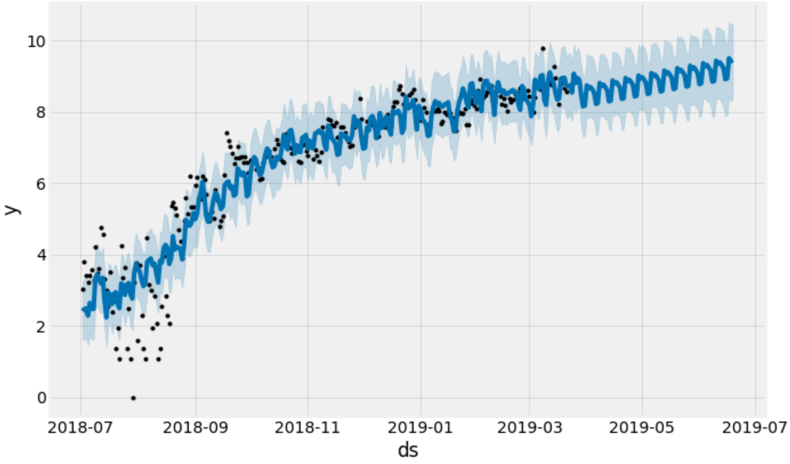


Fig 5.

IV. RESULTS AND DISCUSSIONS

In this work, price values for the crops have been analyzed and presented in the figure below. In this work, the dataset of the crop has been considered and taken from the year 2012 to 2021.

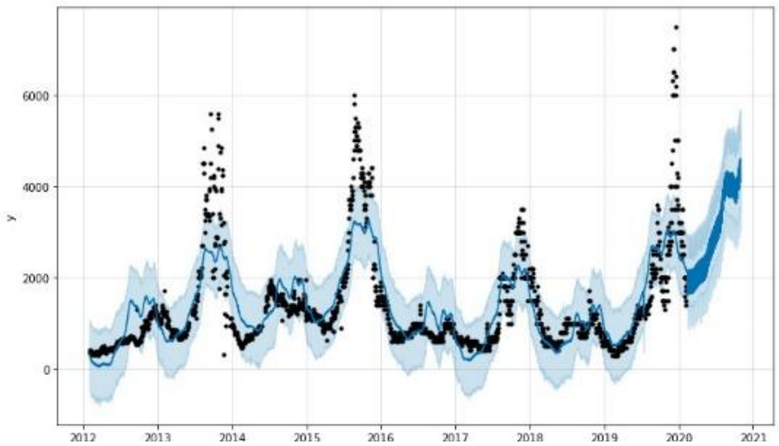


Fig 6. Price Variation of the crop from 2012 to 2021

In the above figure Fig7, the variation is shown by plotting graph datastamp vs the price of the crop(Rs)/quintal. The actual values are shown till the year 2020 and the further procedure of prediction is done for the period of 2020 to 2021.

In fig.8, daily, weekly and yearly trends of the crop prices are shown through the graph. Infig.2, daily,weekly and yearly trends of the crop prices are shown through the graph.

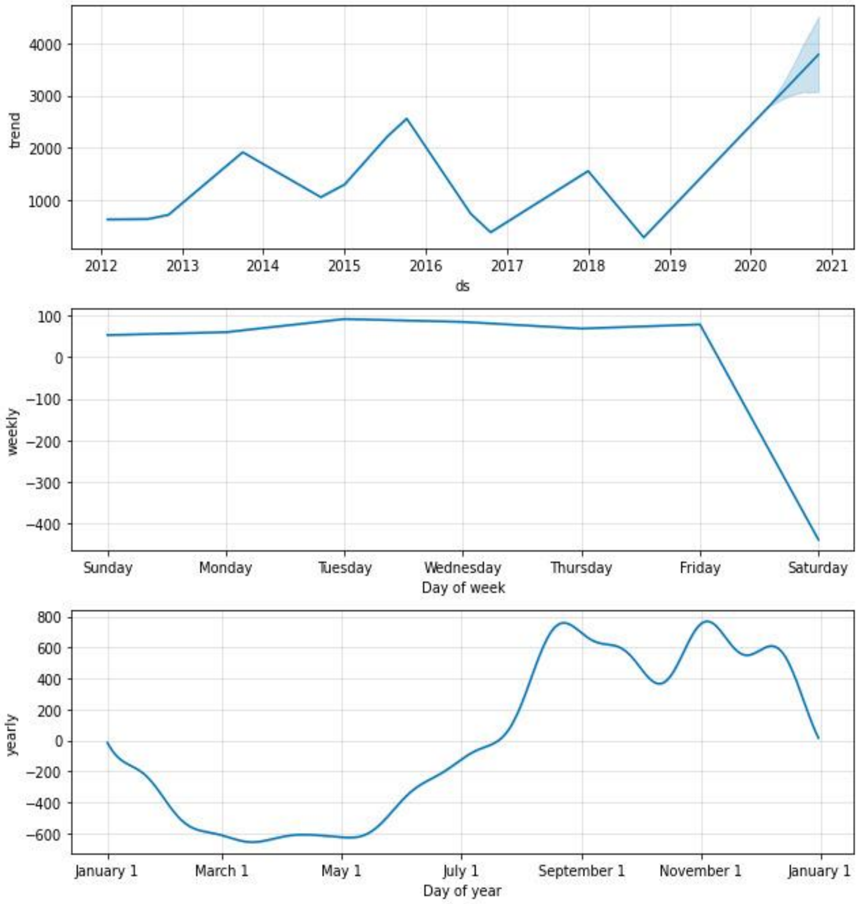


Fig 7.

V. CONCLUSION AND FUTURE WORK

Thus implementing the above procedure would help the farmer grow financially and earn profit in efficient way. This will ultimately lead to decrease in loss faced by famers and help them to know the market condition in a simpler way.

REFERENCES

[1] B V, & .M, Dakshayini. (2017). Demand-prediction model for forecasting AGRI-needs of the society. 430-435. 10.1109/ICICI.2017.8365388.

[2] M. Elferink and F. Schierhorn (2016) Global demand for food is rising Can we meet it?. Harv. Bus. Rev.doi 10.1111/agec.12089

[3] N. Alexandratos and J. Bruinsma (2012) World agriculture towards 2013/2050: the 2012 revision. ESA Work. Pap. No. 12-03, no. 12.

[4] HuyVuong Nguyen, M.Asif Naeem, NuttananWichitaksorn, and Russel Pears. “A smart system for short-term price prediction using time series models”, Computers and Electrical Engineering, 76, pp.339–352, 2019

[5] Kaur, Manpreet & Gulati, Heena &Kundra, Harish. (2014). Data Mining in Agriculture on Crop Price Prediction: Techniques and Applications. International Journal of Computer Applications. 99. 1-3. 10.5120/17422-8273.

[6] S. Rajeswari, K. Suthendran.(2019).Developing an Agricultural Product Price Prediction Model using HADT Algorithm.International Journal of Engineering and Advanced Technology (IJEAT).

[7] Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2

[8] Parul Pandey (n.d.),Predicting the ‘Future’ with Facebook’s Prophet, Medium, accessed 7th May 2020,https://towardsdatascience.com/predicting-the-future-with-facebook-s-prophet-bdfe11af10ff

[9] Ruan van der Merwe(n.d.),Implementing Facebook Prophet efficiently, Medium, accessed 7th May 2020,https://towardsdatascience.com/implementing-facebook-prophet-efficiently-c241305405a3