Trend Analysis, Visualization and Forecasting of Climate Change factors via Time Series Modelling

J – Component Project Report for the course

CSE4004 – Geographic Information System

By

Sai Sahithi Kodavati (18MIS1096)

Abinaya Sruthi Sreeram (18MIS1028)

Eric Brian Anil (18MIS1030)

Submitted to

Dr. RISHIKESHAN C A

Senior Assistant Professor

SCOPE – VIT Chennai

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT CHENNAI

VANDALUR – KELAMBAKKAM ROAD

CHENNAI – 600 127

DECEMBER 2021

BONAFIDE CERTIFICATE

Certified that this project report entitled "**Trend Analysis, Visualization and Forecasting of Climate Change factors via Time Series Modelling"** is a Bonafide work of **Sai Sahithi Kodavati (18MIS1096), Abinaya Sruthi Sreeram (18MIS1028) & Eric Brian Anil (18MIS1030)** who carried out the Project work under my supervision and guidance.

Dr. RISHIKESAN C.A
Associate Professor Senior
School of Computer Science and Engineering (SCOPE),
VIT Chennai, Chennai – 600 127.

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, our course faculty, **Dr. Rishikeshan C A**, Senior Assistant Professor, School of Computer Science and Engineering (SCOPE), VIT Chennai, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

TABLE OF CONTENTS

ABSTRACT

One of the biggest problems that has been continuously riddling our generation for years has been the gradual but very prominent strides of destruction caused by climate change. From melting glaciers to forest fires, there has been multiple indicators over time suggesting that we're exhaustively worsening the condition of the planet for the worse.

Rise in temperatures, Excessive carbon footprint, Concentration of greenhouse gas emissions, etc. have been some of the very clear indicators of climate change indirectly or mostly directly influenced by human activities. The 1.5 degree temperature goal of the Paris Agreement to cut down global warming has been mutated by multiple countries due to their excessive industrial emissions as well. Yet despite this, there has been unnerving debates on the legitimacy of climate change, with focus groups arguing that it is a political hoax.

An analysis of factors such as temperature variations, carbon / greenhouse emissions, etc. and other factors leading to global warming, would provide a documented insight on where each locality stands on the basis of their temperature and emissions and how it has changed over time.

Furthermore, this baseline (historic / past) time variant datapoints of the factors could be used to do a univariate/ multivariate Time Series Forecast to predict the potential temperature changes of each of these locations to 'n' months / years into the future based on changes till this point. The use of machine learning these would also help us to find trends in the change in temperature caused over time and what factor of each location has affected their change.

This study would help in categorising locations into multiple zones based on their climate health levels. The predictions would also help to classify the regions that are in immediate danger of worsening climate scenarios in the near future. Climate related data is usually stored in scientific file formats that require specialized software and can seem unintelligible to those unfamiliar with climate terms and concepts. We look forward to map and project these together to provide a visually aiding insight that's understandable, while trying to understand major climate concepts and familiarise ourselves with real life data in regard to these.

INTRODUCTION AND PROJECT DESCRIPTION

The primary aim behind this project is to visualize and analyze factors such as temperature changes, variation in precipitation rates, etc. across various locations over a time period to see the impact caused by climate change in multiple locations. This time-series data could then be used to predict future forecasts of the same attributes in places around the world. These future forecasts can then be visualized again in various 2d-3d map / scene forms to make it easier to perceive and understand the changing effects of the values in these places for comparison.

Our project aims to do an analysis of multiple locations using a GIS software, preferably with our locality to be set in India for starters with bound to expansion as we proceed. The analysis would take in factors such as temperature variations, carbon / greenhouse emissions, etc. and other factors leading to global warming, and sketch out the changes over time in various localities using an ArcGIS software. This would provide a documented insight as per our goal, to showcase where each locality stands on the basis of their temperature and emissions and how it has changed over time.

Furthermore, these documented insights could be used to forecast future values of the undertaken attributes so as to see the trend in climatic changes over various localities in time and predict what they turn out to be in the future.

In short, our target output for this project is to generate a case study regarding climate change trends based on a few relevant attributes and to experiment with a model that could be opensource as a base geographical document for people to add on data and forecast changing climatic variables locally and on a global scale to bring about awareness regarding the factors - small or big that causes variations in the climatic scenario and to pinpoint probable causes for ease in representation and understanding.

TOOLS AND TECHNICAL STACK

As for our GIS Tool for geoprocessing and handling of data, mapping, plotting and visualizations, etc., we've selected **ArcGIS Pro 2.8** in the Windows OS as our preferred software of choice, for the massive commercial functionalities it provides as compared to similar tools. Beyond the functionalities, they also have a well developed community and a massive amount of educational resources and tools for students and beginners, for a smooth onboarding towards even the bleeding edge and critical of features while making it easy to familiarize yourself with the toolsets. For the duration of the project, we have used the 3 month Educational Trial Plan available for students and teachers, enabling access towards the primary features of ArcGIS Pro and ArcGIS Online, alongside ArcMap. Initial data preprocessing however, was done within Microsoft Excel.

As for the forecasting part, of the two modules present – the programming has been carried over in **Python 3.8** as our preferred language for computation. The forecasting has been executed in a **Google Colab** notebook for better collaboration and ease in accessing advanced computational resources for the Machine Learning part of the project.

The essential packages / libraries that we have used in Python for the time-series forecasting are as follows :

- Pandas (Data Handling / Pre Processing)
- Numpy (Multidimensional Numerical Computation)
- Scipy (Scientific computing and complec mathematical modules)
- Statsmodels.api (Simulation and Estimation of Statistical Models)
- Matplotlib (Plotting / Visualisation Library)

MODULES

The key modules involved in this case study to create correlation between the various factors were:

1. **Data Collection & Processing**
2. **Map Creation & Basic Visualization**
3. **Data Transformation: Static Data to Temporal Data & Temporal Data Projection**
4. **Conversion to Multidimensional Data & Space Time Cube Representation**
5. **Time Series Forecasting – Methodologies, Comparison and Prediction**
6. **Analysis of the Forecast and further processing (Result and Future Scope)**

DATA COLLECTION & PROCESSING

For our initial experimentation during the time bracket of this J Component, we decided to restrict our consideration for the climate data of India, before any scale ups. Our choices were between choosing national temperature averages for a year globally or to stick to a smaller perimeter. Setting state political boundaries felt the most appropriate for the project to get the right insights as, a larger national scale of data would give misleading predictions due to unequal aggregation of data at various places and any smaller / granulated division such as district boundaries would not just be a tedious task in terms of data regulation, but also in collection. Since most climate related datasets were present in varying formats, had major inconsistencies within and about the values and attributes, it felt impossible within the time frame to be able to create a model that would be rightly accustomed to district boundaries taking into consideration these anomalies (For example, in a state like Kerala most datasets would only have details, albeit irregular, of 2-3 out of the total 14 districts). Hence we fixated on choosing state boundaries in India as our locality division when considering the climate factors.

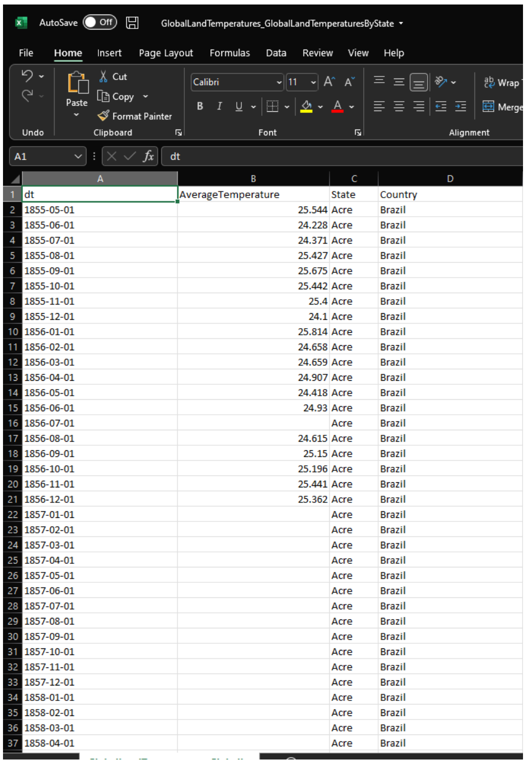
Now the next crucial step was deciding and acquiring what kind of attributes we would be choosing to measure the variations in climate. As mentioned earlier there are numerous factors such as temperature variations, carbon / greenhouse emissions, change in precipitation, etc. that contributes to the current changing climate scenario. Of these our aim was to find any 1-2 attribute and try to document the variation amongst these over time in various localities – and to predict how these patterns may indicate what's in future for these localities, what places are bound to drastic changes, and what kind of changes.

Considering all the everchanging attributes, we chose temperature rise and precipitation variations as our key attributes to explore for the project. 2020 was reported to be the hottest year in over a century and has witnessed irregular rainfall patterns according to various reports. These stood out to us as factors that could, even in small changes, determine massive global impact. Hence, the two primary time variant data we've collected are of Annual Temperature variation in degree Celsius and Annual Precipitation rates in mm of various states.

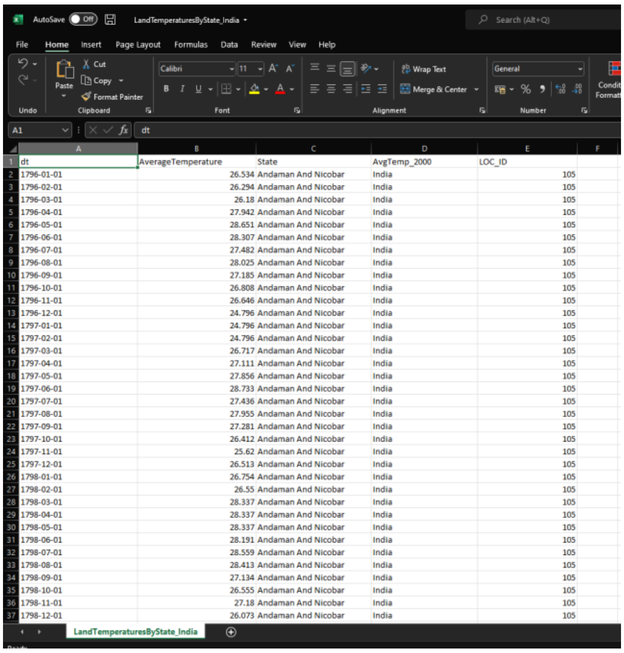
Procurement of Data was the next step – in various types, formats, from various sources, etc. with constant focus on the verifiability of it.

To start off for the Indian Administrative Map, the shapefile for the spatial data of Indian map (as bounded by administrative borders of regions, states, etc. for ease in polygons) was taken from **diva-gis.org** and **gisclimatechange.ucar.edu**, both research based sources collecting geographic data for the same purpose over time. Out of the multiple shapefiles acquired, we chose the one with state political boundaries to be our base shapefile.

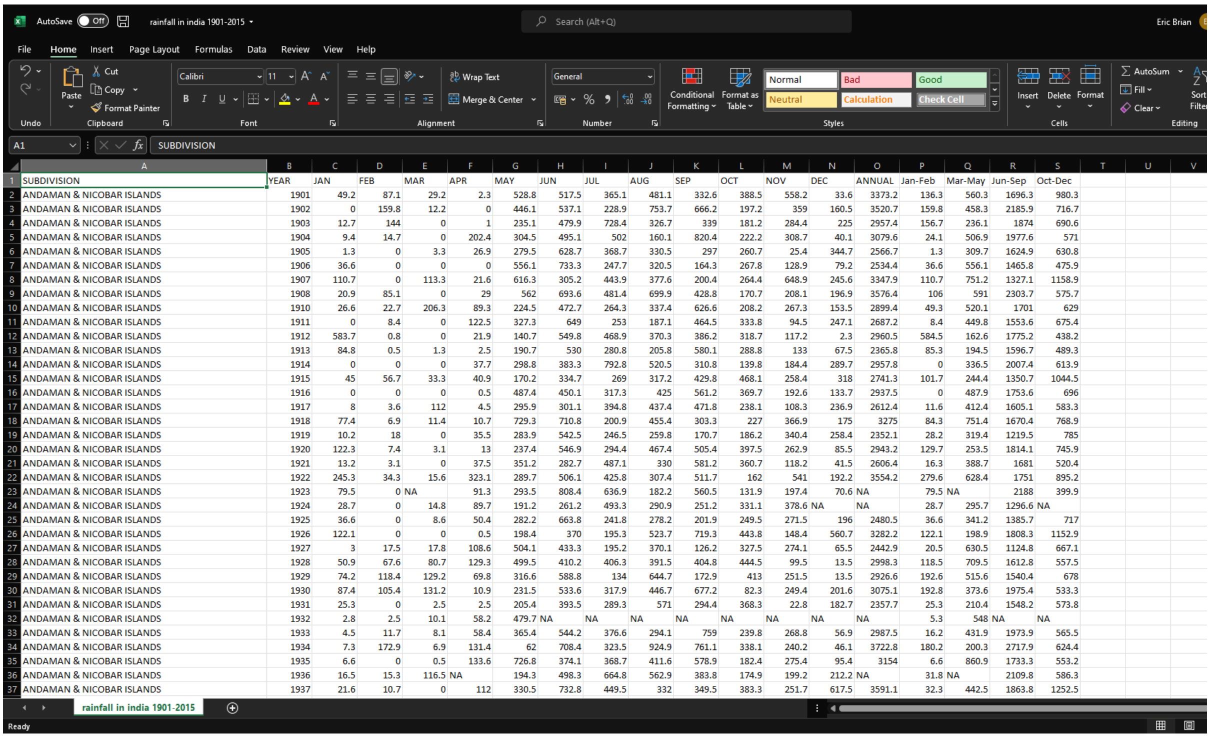
The collection and processing of datasets were the confusing part. As mentioned, the variety of data available was never consistent – be it in terms of format or attributes or values without anomalies. However after juggling through multiple sources, we collated multiple datasets based on their susceptibility. The climate related datasets (regarding pollution, carbon footprint, temperature and precipitation – both global and local) were taken from **data.world**, **data.gov.in**, **cdsp.imdpune.gov.in**, **library.noaa.gov** and **climateknowledgeportal.worldbank.org.** These included repositories of geographic datasets compiled by organizations, research institutes and governmental institutions for their research purposes. The stray data had measurements for a time period of about 1850s to late 2010s till 2020. Out of these we selected the CSV ones featuring data regarding monthly Temperature and Precipitation for India as split by states.



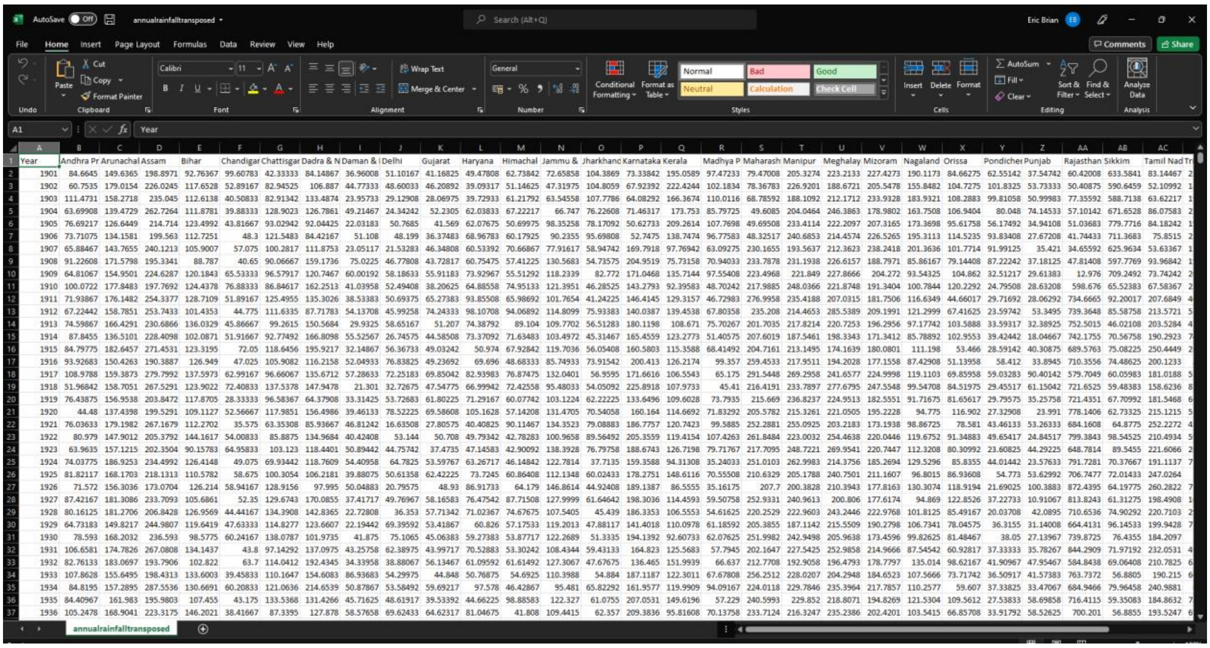
dt	AverageTemperature	State	Country
1855-05-01	25.544	Acre	Brazil
1855-06-01	24.228	Acre	Brazil
1855-07-01	24.371	Acre	Brazil
1855-08-01	25.427	Acre	Brazil
1855-09-01	25.675	Acre	Brazil
1855-10-01	25.442	Acre	Brazil
1855-11-01	25.4	Acre	Brazil
1855-12-01	24.1	Acre	Brazil
1856-01-01	25.814	Acre	Brazil
1856-02-01	24.658	Acre	Brazil
1856-03-01	24.659	Acre	Brazil
1856-04-01	24.907	Acre	Brazil
1856-05-01	24.418	Acre	Brazil
1856-06-01	24.93	Acre	Brazil
1856-07-01		Acre	Brazil
1856-08-01	24.615	Acre	Brazil
1856-09-01	25.15	Acre	Brazil
1856-10-01	25.196	Acre	Brazil
1856-11-01	25.441	Acre	Brazil
1856-12-01	25.362	Acre	Brazil
1857-01-01		Acre	Brazil
1857-02-01		Acre	Brazil
1857-03-01		Acre	Brazil
1857-04-01		Acre	Brazil
1857-05-01		Acre	Brazil
1857-06-01		Acre	Brazil
1857-07-01		Acre	Brazil
1857-08-01		Acre	Brazil
1857-09-01		Acre	Brazil
1857-10-01		Acre	Brazil
1857-11-01		Acre	Brazil
1857-12-01		Acre	Brazil
1858-01-01		Acre	Brazil
1858-02-01		Acre	Brazil
1858-03-01		Acre	Brazil
1858-04-01		Acre	Brazil



dt	AverageTemperature	State	AvgTemp_2000	LOC_ID
1796-01-01	26.534	Andaman And Nicobar	India	105
1796-02-01	26.294	Andaman And Nicobar	India	105
1796-03-01	26.18	Andaman And Nicobar	India	105
1796-04-01	27.942	Andaman And Nicobar	India	105
1796-05-01	28.651	Andaman And Nicobar	India	105
1796-06-01	28.307	Andaman And Nicobar	India	105
1796-07-01	27.482	Andaman And Nicobar	India	105
1796-08-01	28.025	Andaman And Nicobar	India	105
1796-09-01	27.185	Andaman And Nicobar	India	105
1796-10-01	26.808	Andaman And Nicobar	India	105
1796-11-01	26.646	Andaman And Nicobar	India	105
1796-12-01	24.796	Andaman And Nicobar	India	105
1797-01-01	24.796	Andaman And Nicobar	India	105
1797-02-01	24.796	Andaman And Nicobar	India	105
1797-03-01	26.717	Andaman And Nicobar	India	105
1797-04-01	27.111	Andaman And Nicobar	India	105
1797-05-01	27.856	Andaman And Nicobar	India	105
1797-06-01	28.733	Andaman And Nicobar	India	105
1797-07-01	27.436	Andaman And Nicobar	India	105
1797-08-01	27.955	Andaman And Nicobar	India	105
1797-09-01	27.281	Andaman And Nicobar	India	105
1797-10-01	26.412	Andaman And Nicobar	India	105
1797-11-01	25.62	Andaman And Nicobar	India	105
1797-12-01	26.513	Andaman And Nicobar	India	105
1798-01-01	26.754	Andaman And Nicobar	India	105
1798-02-01	26.55	Andaman And Nicobar	India	105
1798-03-01	28.337	Andaman And Nicobar	India	105
1798-04-01	28.337	Andaman And Nicobar	India	105
1798-05-01	28.337	Andaman And Nicobar	India	105
1798-06-01	28.191	Andaman And Nicobar	India	105
1798-07-01	28.559	Andaman And Nicobar	India	105
1798-08-01	28.413	Andaman And Nicobar	India	105
1798-09-01	27.134	Andaman And Nicobar	India	105
1798-10-01	26.555	Andaman And Nicobar	India	105
1798-11-01	27.18	Andaman And Nicobar	India	105
1798-12-01	26.073	Andaman And Nicobar	India	105



The biggest challenge following this was the **Data Cleaning** and Processing involved, and confirming on a format required from all the available data. We chose to move forward with datasets having Temperature as our first attribute of preference for testing.

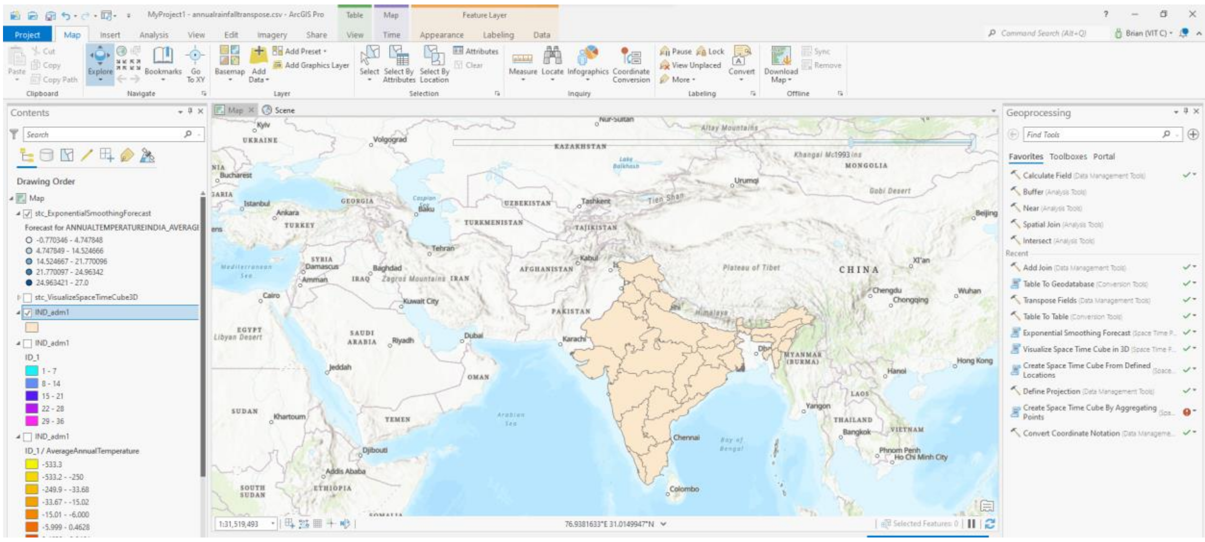


Most datasets had monthly values from around the late 1800s till the early 2010s on an average. This meant that we had about 10 lakh rows of data points to handle and organize.

After doing the basic cleansing via dropping the NA / non numerical columns, maintaining a consistent format in columns (For ex: A date value could have anomalies by having multiple formats within the same file as yyyy-mm-dd and mm-dd-yy), and calculating the average of the missing values to normalize each dataset that we had selected, we proceeded to merge values between similar datasets to maintain consistency in the locality and dates. This was done because some datasets might have had 28 states while another would have had 27 or 29, or some would have monthly values for all the 12 months, while a few would have been quarterly measurements. Hence we worked on normalizing these for an acquired format, before moving it to ArcGIS for mapping.

MAP CREATION & BASIC VISUALIZATION

Post cleansing and processing the data to maintain a consistent format, it was time for it to be used in ArcGIS for mapping. After familiarizing ourselves with the basics of all the 3 variants of the commercial tool – ArcGIS Online, ArcMap and ArcGIS Pro – we proceeded to create a map based project in ArcGIS Pro.



After the setup of the geodatabase of the project, we imported the Indian Map shapefile (titled IND_adm1) that consisted polygons of the administrative state boundaries as of 2019.

Each polygon was mapped based on the state to the base map template of physical hill-based features present in the map. It did not contain any Latitude-Longitude values but had Object IDs resonating to its location and area on the plane. It also used the **WGS Geographic coordinate system** for representation.

Each state was mentioned alongside its attributes as type, which were viewable in the attribute table option. The naming conventions for the states were also defined in the table.

	FID	Shape *	ID_0	ID_1	NAME_1	ENGTYPE_1
1	0	Polygon	105	1	Andaman and Nicobar	Union Territory
2	1	Polygon	105	2	Andhra Pradesh	State
3	2	Polygon	105	3	Arunachal Pradesh	State
4	3	Polygon	105	4	Assam	State
5	4	Polygon	105	5	Bihar	State
6	5	Polygon	105	6	Chandigarh	Union Territory
7	6	Polygon	105	7	Chhattisgarh	State
8	7	Polygon	105	8	Dadra and Nagar Hav...	Union Territory
9	8	Polygon	105	9	Daman and Diu	Union Territory
10	9	Polygon	105	10	Delhi	Union Territory
11	10	Polygon	105	11	Goa	State
12	11	Polygon	105	12	Gujarat	State
13	12	Polygon	105	13	Haryana	State
14	13	Polygon	105	14	Himachal Pradesh	Union Territory
15	14	Polygon	105	15	Jammu and Kashmir	State
16	15	Polygon	105	16	Jharkhand	State
17	16	Polygon	105	17	Karnataka	State
18	17	Polygon	105	18	Kerala	State
19	18	Polygon	105	19	Lakshadweep	Union Territory
20	19	Polygon	105	20	Madhya Pradesh	State
21	20	Polygon	105	21	Maharashtra	State

0 of 36 selected

Now it was time to add the modified, cleaned dataset regarding the state based monthly temperatures for display. It was imported as a csv data table first.

	Year	India	Andaman and Nicobar	Andhra Pradesh	Assam	Delhi	Goa	Gujarat	Haryana	Himachal Pradesh	Karnataka	Kerala	Lakshadweep	Maharashtra	Manipur	Meghalaya	Mizoram	Nagaland	Orissa
1	1901	24.26	27.81	27.04	22.77	25.27	26.26	26.66	25.09	9.8	25.44	25.89	27.12	26.17	19.2	21.37	21.11	18.69	25.41
2	1902	24.52	27.84	27.36	22.5	25.52	26.64	27.19	25.34	10.01	25.75	25.92	27.04	26.74	18.92	21.18	20.9	18.34	25.6
3	1903	23.96	27.63	26.77	22.73	24.9	25.96	26.2	24.68	9.4	25.18	25.72	27.01	25.86	19.17	21.36	21.09	18.63	25.25
4	1904	24.02	27.29	26.83	22.51	24.95	26.07	26.59	24.77	9.57	25.22	25.46	26.62	26.08	18.96	21.14	20.87	18.42	25.16
5	1905	23.87	27.8	26.96	22.11	24.65	26.13	26.25	24.43	9.03	25.38	25.83	27.07	25.99	18.57	20.72	20.49	18.02	25.03
6	1906	24.11	27.92	27.15	22.64	24.87	26.28	26.31	24.69	9.52	25.51	25.93	27.14	26.13	19.12	21.27	21.05	18.55	25.4
7	1907	23.97	27.31	26.92	22.52	24.66	26.18	26.42	24.48	9.25	25.34	25.65	26.82	26.11	18.97	21.14	20.91	18.42	25.21
8	1908	24.01	27.24	26.85	22.86	24.84	26.02	26.1	24.62	9.53	25.22	25.58	26.77	25.96	19.27	21.47	21.17	18.75	25.36
9	1909	23.92	27.25	26.92	22.97	24.41	26.16	26.09	24.22	9.14	25.33	25.62	26.81	26.04	19.41	21.54	21.3	18.91	25.21
10	1910	23.8	27.2	26.75	22.51	24.5	25.96	26.05	24.34	9.19	25.16	25.56	26.78	25.87	18.94	21.1	20.84	18.42	25.07
11	1911	24.14	27.45	27.12	22.4	24.91	26.43	26.52	24.74	9.5	25.59	25.92	27.06	26.34	18.84	21.04	20.76	18.29	25.27
12	1912	24.29	27.45	27.32	22.54	24.93	26.58	26.84	24.74	9.51	25.76	26	27.12	26.58	19.02	21.18	20.96	18.44	25.45
13	1913	23.98	27.51	27.13	22.35	24.59	26.33	26.41	24.4	9.12	25.54	25.88	27.07	26.22	18.83	20.96	20.75	18.28	25.16
14	1914	24.1	27.52	27.01	22.47	24.77	26.36	26.63	24.6	9.4	25.53	25.92	27.17	26.28	18.91	21.08	20.82	18.37	25.2
15	1915	24.5	27.54	27.32	23.09	25.39	26.51	26.89	25.19	10	25.71	26.06	27.26	26.51	19.53	21.7	21.44	19	25.65
16	1916	24.12	27.2	26.97	22.96	24.82	26.25	26.54	24.72	9.67	25.4	25.72	26.92	26.2	19.42	21.51	21.27	18.93	25.28
17	1917	23.47	27.11	26.43	22.58	23.85	25.73	25.93	23.7	8.74	24.89	25.37	26.66	25.56	19.06	21.11	20.91	18.56	24.75
18	1918	24.06	27.24	26.95	22.57	24.89	26.31	26.53	24.7	9.43	25.39	25.6	26.76	26.27	19.03	21.11	20.89	18.53	25.16
19	1919	24.1	27.77	27.24	23.06	24.52	26.46	26.48	24.34	9.1	25.65	26.01	27.18	26.3	19.54	21.58	21.38	19.05	25.39
20	1920	24.28	27.31	27.4	22.96	24.78	26.53	26.69	24.5	9.15	25.72	26.02	27.13	26.53	19.42	21.55	21.3	18.9	25.68
21	1921	24.54	27.26	27.23	22.96	25.69	26.4	27	25.44	10.07	25.56	25.79	27.03	26.55	19.44	21.53	21.27	18.96	25.68

0 of 120 selected

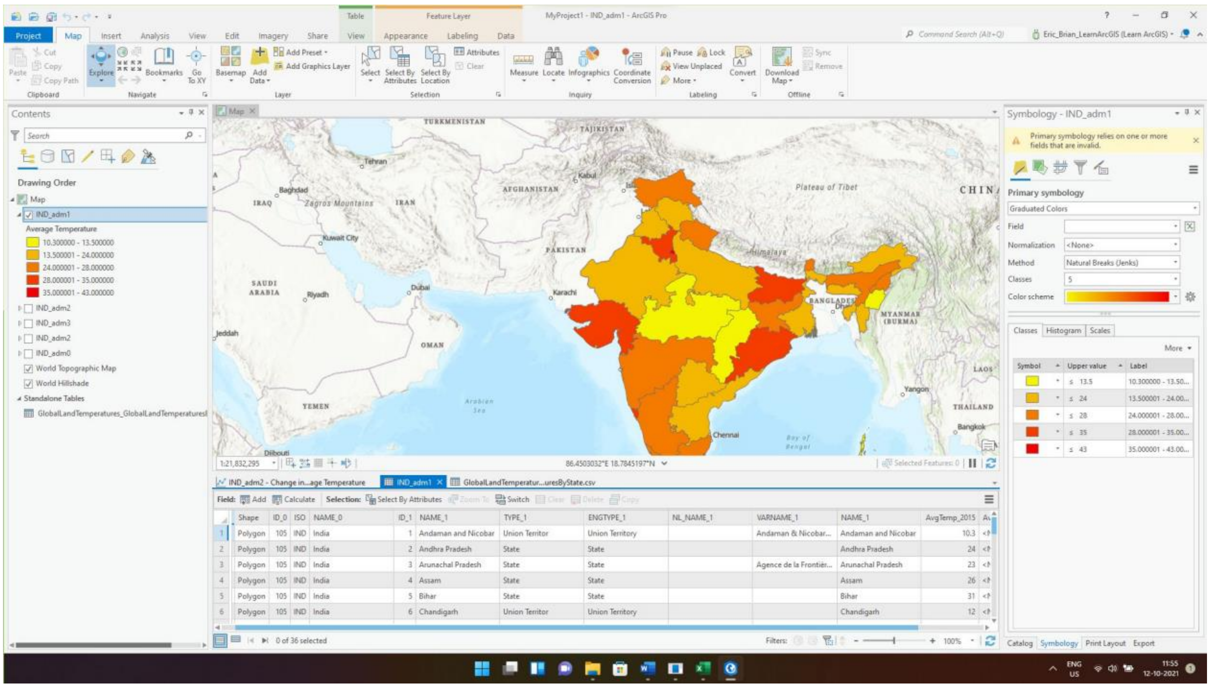
It is clear that the format of the Temperature dataset and that of the attribute table of the shapefile are noticeably different – right from the naming conventions used to the attribute format. This is where we started exploring the data handling tools present within ArcGIS to manage attributes and values to fit formats.

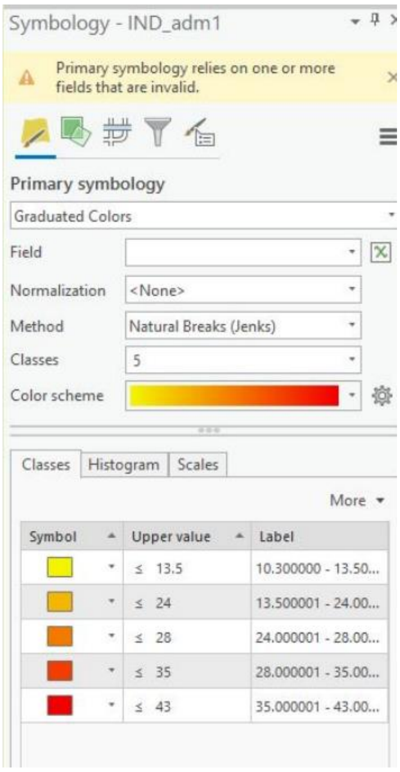
Once we fixed the Attribute values and the remaining anomalies it was time to merge the attributes so as to get a proper representation of the current data. The dataset had been modified to contain monthly data from 1980 to 2000 initially to keep the dataset as light as possible.

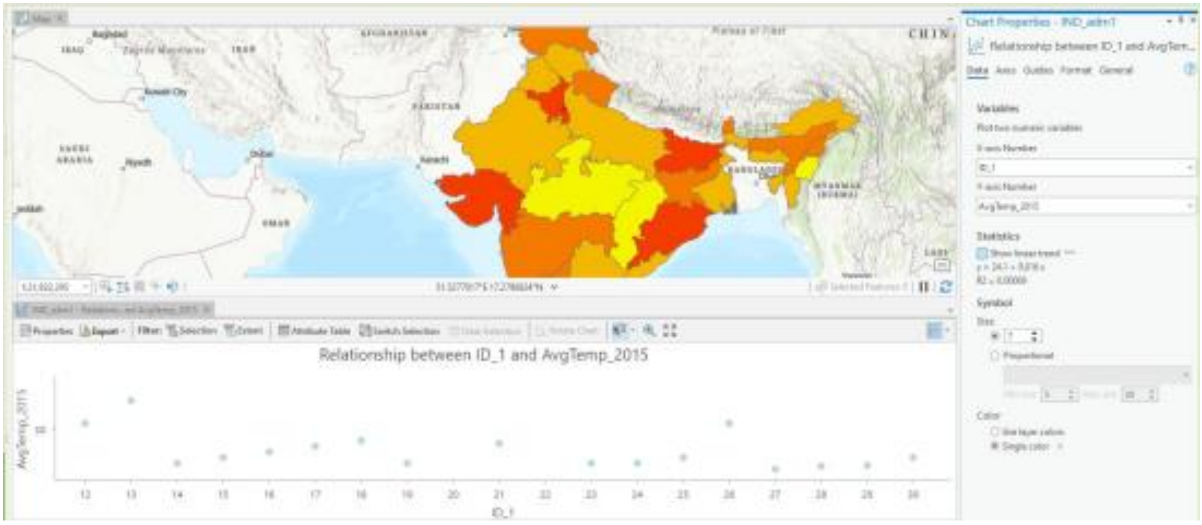
The merging of both these tables could be done by the **Join Table** function present between the attributes of the shapefile and the csv file in ArcGIS, where it automatically converts the current table to detect relations between attributes of both the tables and merge them based on the parameters we set. Giving the Name_1 as the parameter on the shapefile, and the States as the parameter on the CSV table, the Join Table function client detected 1-to-1 relations between the tables and proceeded to match and merge.

Once the tables are merged, it is easier to view the temperature values of a year now. It is visible as attribute table values, bur for better visibility in visualization, we could view it using the **Graduated Gradients** option in the **Symbology** tab of ArcGIS.

Within this tab, you could create the number of categories you like depending on the range of the Temperature, set a Color gradient, select the required attribute parameter on the table and go ahead to visualize the gradual change of the Temperature of a place over multiple states in a month now.





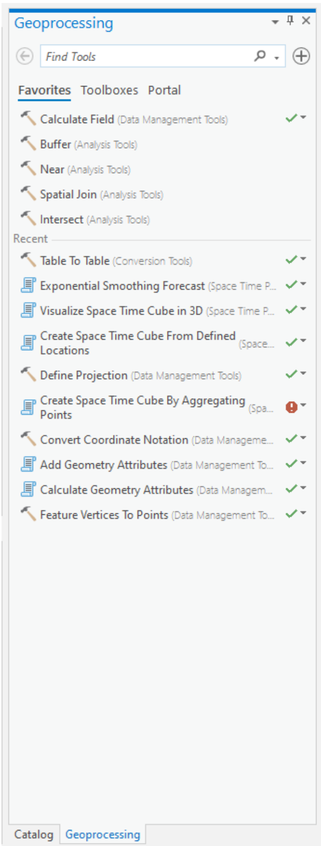


This could be used to chart the temperature variations over different places as well. The first stage of our visualizations is done. However, looking at the Attribute Table of the shapefile, you could realise that it has only acquired data points in a state in regards to one specific timeframe, i.e, of one month. So to view changes over a specific place in different points of time (different months, years, etc), you'd have to repeat this process in the shapefile for each time point as different layers in the shapefile. This could easily be hectic as we have not converted this in the required format for Time Series Visualization yet, as Temporal data.

DATA TRANSFORMATION: STATIC DATA TO TEMPORAL DATA & TEMPORAL DATA PROJECTION

Temporal Data is nothing but data that has the extra dimension of Time added to its visibility, helping it fluctuate and reflect with changes over time. This required additional data transformation for ease and this is where we started off for work since our Review 2.

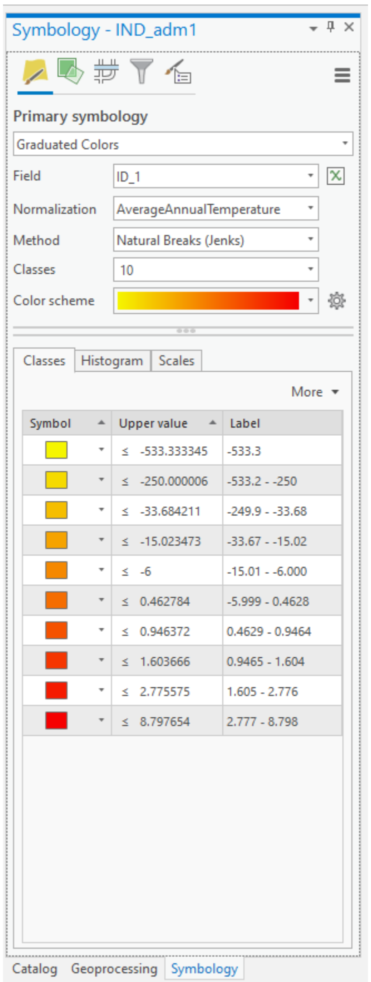
Our first move was in aggregating the data points we had to convert it from a monthly temperature to a yearly one. This involved converting all the monthly values of a year under each state attribute by taking the mean of it all to be reflected as a single row of temperature measurement for a year. This time throughout, we reinforced the power of the **Geoprocessing Toolset** available within ArcGIS for almost every major operation, right from ones involving data manipulation, transformation and modification. Aggregation by mean of the temperature from 1901 to 2020 vastly trimmed the data points we had to around 12,000 rows of data.



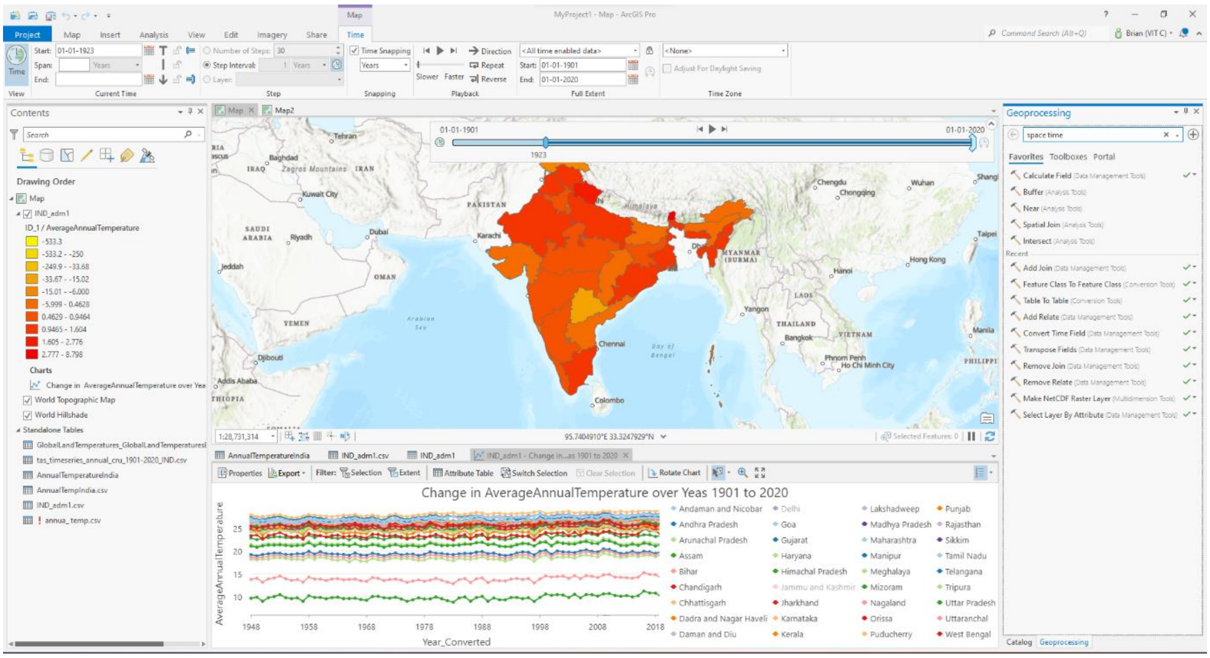
After aggregating average, we realized that the dimension of the data is one of the factors that kept from multiple time points being reflected on the table join with the shape file. Hence, the table having states as the attribute column and years a the rows were **transposed** keeping the year as the key, within ArcGIS itself.

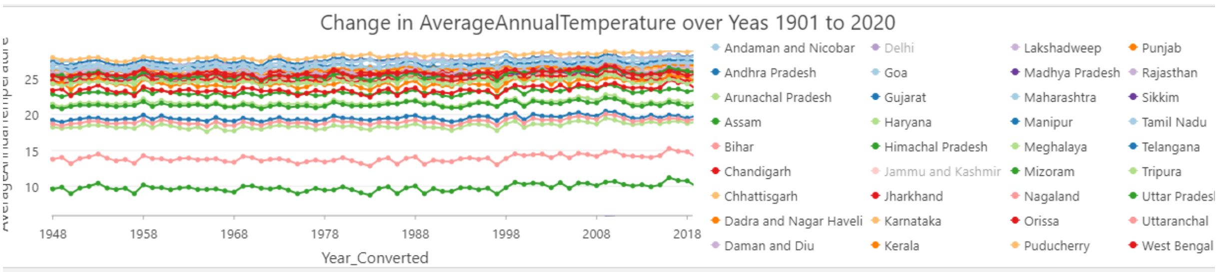
Once this was done, the table was added to the same geodatabase as the shapefile and the project converted from a csv attribute dataset to a generic readable table format within ArcgGIS (This was one of the reasons why previously the join table feature could only find 1-to-1 connections between the attribute tables of the shapefile and that of the temperature dataset).

Furthermore after correcting the date format (The format was in yyyy, but for temporal data representation within ArcGIS, it requires it to be in the format of yyyy-mm-dd. This was done automatically with the help of the Convert DateTimeField feature in the Geoprocessing tool), the tables were once again joined. This time, the Join Table feature client detected 1-to-m relation between the tables and hence created all the 12,000 data points within the software. After using the geoprocessing tab for temporal conversion, the job was done and the only thing left was for live visualization.



Once again, the Graduated Colors visualization of the **Symbology** function was applied with given categories of temperature classes, but this time to a temporal data set with time as the key variant.





As it is visible from the visuals, this time around there appeared a temporal slider as an overlay to the map. This shows all the data variations over the 119 years, loaded dynamically. As we slide through the years, the graduated colors modify themselves in the map to represent the change in the temperature.

The charting provided vast insights as well. Now all the different states, including a national average could be seen plotted as a variation of time over the years within the same chart, which could be identified based on the legend. Even though the variations over a span look very minimal, it is to be noted that since we aggregated the data to have the average of every year, the charting does not show a real-time temperature value, but a **gradual global rise in the temperature average over the span of these years. Alarming but true.**

CONVERSION TO MULTIDIMENSIONAL DATA & SPACE TIME CUBE REPRESENTATION

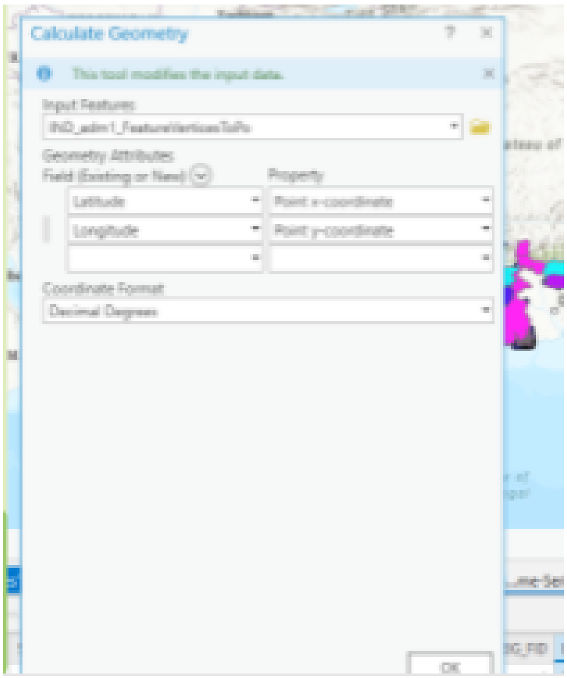
Now, since our temporal data has an added dimension of time to it, storing it any normal 2D raster format will not allow it to be used as a compatible format for computation in any of the procedures we want to follow up with.

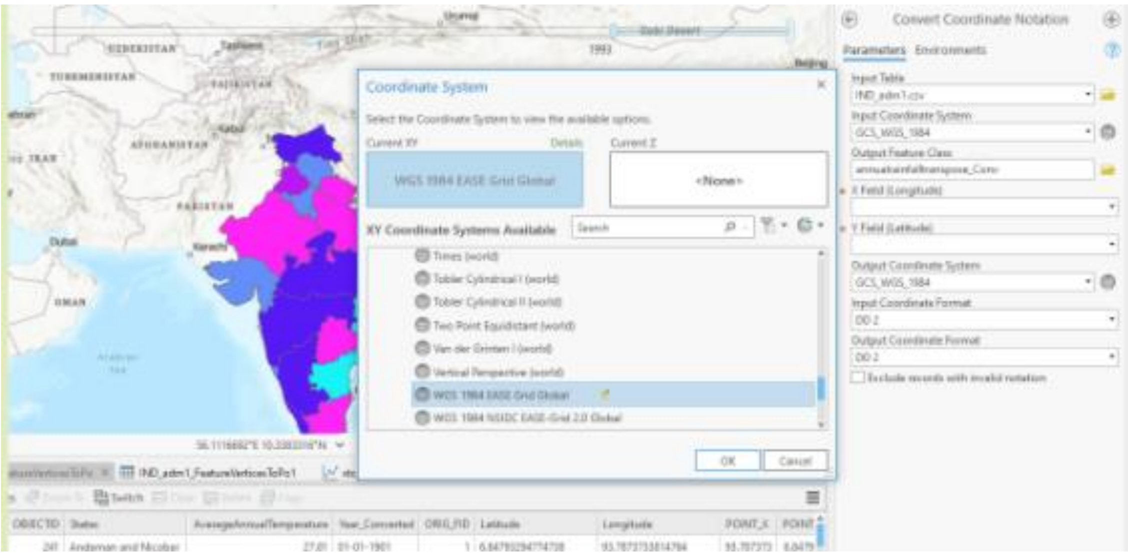
Hence, the current data, with all its values in place had to be converted to a NetCDF (Network Common Data Form) file format. A NetCDF data file is nothing but a multidimensional raster file which is self-describing, machine dependent and able to accommodate the additional temporal dimension to it in its format in any future processing.

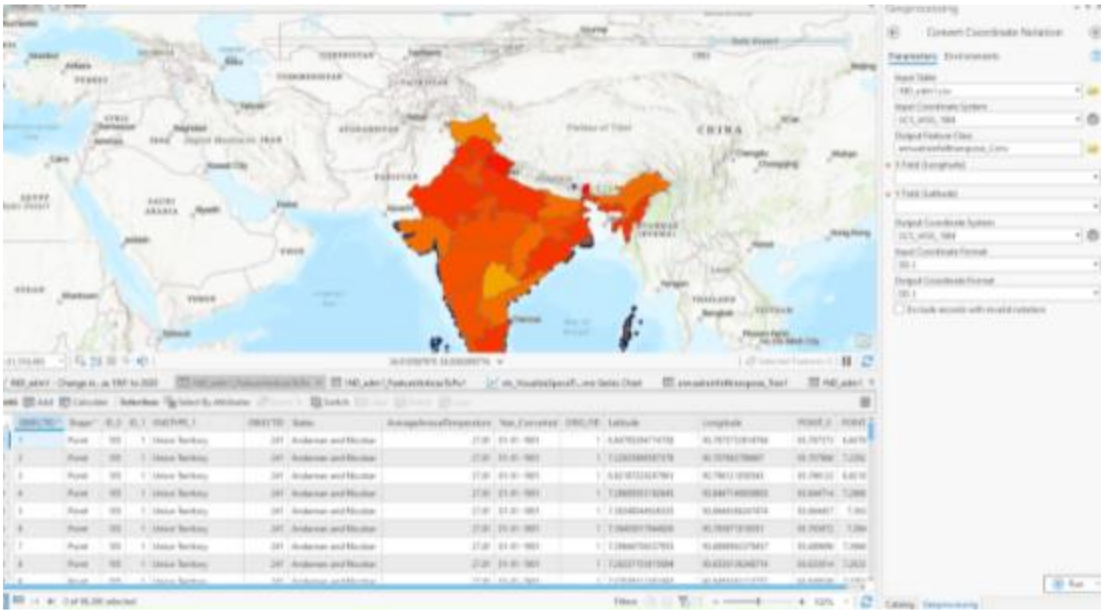
Now converting a Time based data to a NetCDF format is based on locational vector points that corresponded to temporal data. Now, as we mentioned in the start, our shapefile **does not have** any latitude longitude values attached to it and a NetCDF conversion requires a **Projected Coordinate System** as well.

Hence, the next step involved converting the Geographic Coordinate System that existed in the map to a Projected Coordinate System such as the **WSG84** from 1984.

ArcGIS's attribute table geoprocessing tools came in handy once again as the **Calculate Geometry property** for location based tables automatically computed X,Y Lat-Long values for each state that had the polygon object IDs and gave us the point vector for the central point of each state.



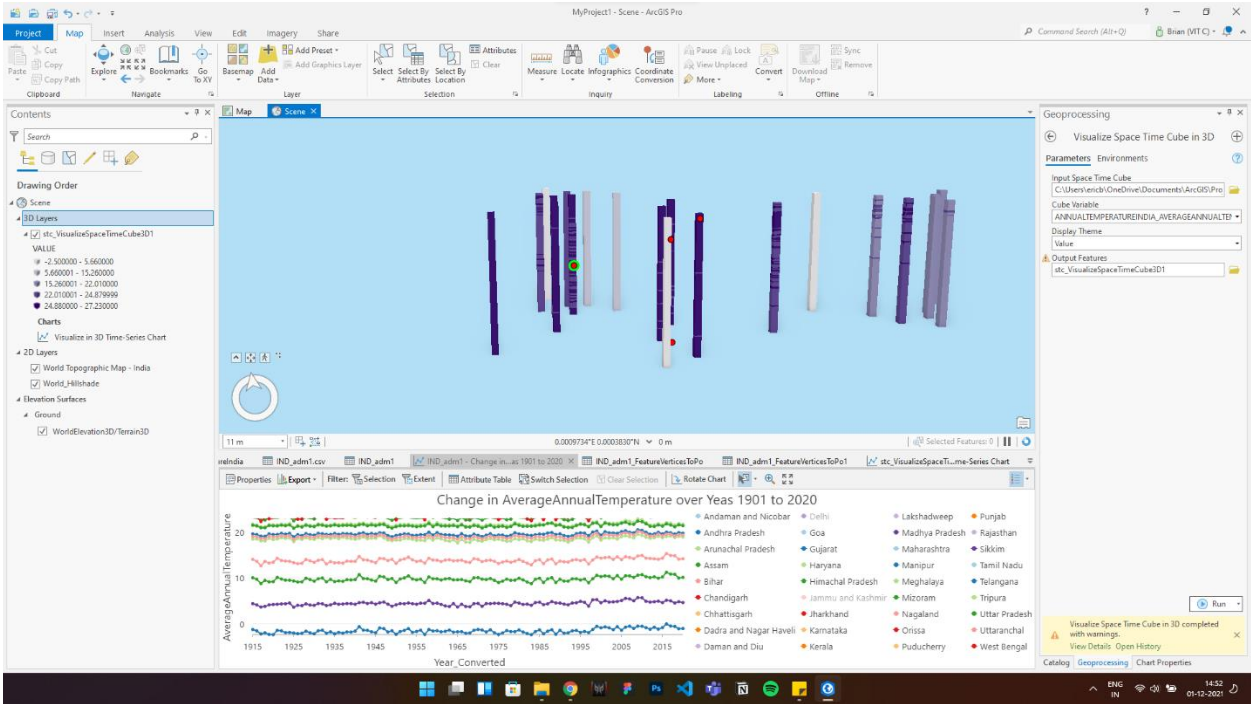




Once these were prepared it was time to convert our data format into a NetCDF format. There were many ways to do it, but functionally since we were aiming towards 3D visualization of the temperature data anyway, we decided to create the format by creating a **3D Space-Time Cube.**

As for creating the Space Time Cube, out of the options available in the GeoProcessing tab, we chose the **Create Space Time Cube By Aggregating Points** tool, since we had converted the polygons to point vertices for this purpose.

After specifying the required parameters, setting time as the variant attribute, selecting the time steps (1-year gap in this scenario), we finally generate the space time cube file.



Till now we've been viewing the Map in a normal Map view within ArcGIS, however the Space Time Cube is basically a 3D model projections on the map that is rendered in real time and hence, we open it up in a new **Scene Tab.** This is where the model will be rendered.

As visible from the image, it is filled with multiple cubes / blocks. Each block represents the temperature variation in that state in a particular year as compared to the previous year. Since we have data from 1901 – 2020, there would be around 119 blocks protruding from the locality of each state. Here too, just like in the graduated colors in Symbology, various categorized classes can be set for labeling prior to the display.

There is a small navigator as seen in the bottom left of the Scene display. The mouse can be used to navigate around various angles, move around further in or away from the blocks, etc. in a very interactive manner, just like how one might use Google Street navigation or so, if not more fluidic.

Each point block, on clicking would give you a popup defining the exact details – State, Location, Year, Temperature, Variation from the previous year, etc. for more info. This is hence a very interactive way of representing and understanding data, especially ones that come as a function of time across locations.

(The screenshot was taken while it's still rendering the blocks in 3D as it's a little time consuming process depending on GPU usage. The block structures in white are still rendering their prediction).

TIME SERIES FORECASTING

Time series forecasting was a key module as well as a very important point of learning for this project. Since the idea behind the data collection was to view the trends in the attributes as rainfall and temperature, and to predict possible values for the near future based on the historical analysis of the data – Investigating and finalizing on the most appropriate Time Series modelling procedure required a homework of study.

As mentioned, the data we used for visualization and now for prediction, has been a univariate tabulation of the attributes – which means that at a time, we are only considering any one attribute (be it the temperature, or the precipitation) as a time varying variable for a give location (state in India in this case), and not a combination of multiple factors to find a correlation or create a bias based on that. Hence our first mission was to cement the selection of any one univariate time series model. Something was not as heavy as a deep learning model, and would be easy to handle would fit the purpose of the project.

The most popular amongst univariate models, under the stats models package was the availability of the ARIMA (AutoRegressive Integrated Moving Average) model.

Now for fine-tuning an ARIMA model, the autoregressive variables **p, d** and **q** are crucial factors in determining the learning of trends. In an ARIMA model generally, Y = (Auto-Regressive Parameters) + (Moving Average Parameters). The p and d are autoregressive parameters while q is the moving average model parameter. This is necessary for the model to learn based on trends in the data. But since our dataset was a huge one accumulated over 119 years, anomalies in the consistency on the dataset (besides cleaning) would make it irritably difficult for us to fine tune the parameters and let the model make a relevant prediction.

This is when we researched about a novel model in Time-Series prediction developed by the research team in Facebook known as **FbProphet**, based on the Prophet procedure, which develops on an additive model.

Further data exploration within our GIS Toolset provided us with a new insight about the geo processing toolset consisting on forecasting tools within the ArcGIS Software itself. The 4 main procedures that ArcGIS provided for forecasting temporal data were as follows :

- **Curve Fit Forecast** (uses simple curve fitting to model a time series and forecast future values at every location in a space-time cube.)
- **Evaluate Forecast By Location** (Not a prediction toolset, but used to evaluate and merge multiple forecasts already made of the same underlying time series data at a set of locations)
- **Exponential Smoothing Forecast**
- **Forest based forecast** (uses forest-based regression to forecast future time slices of a space-time cube. The training data used to build the forest regression model is constructed using time windows on each location of the space-time cube.)

Of this, being a univariate data, with a slightly varying linear approach, with certain values showing an exponential rise at times, We figured that the **Exponential Smoothing Forecast** might be the closest fit to a procedure we could experiment with for our purpose. The Curve Fit forecast model might also have made a similar impact, but there were a few advantages in terms of datatype handling that made us choose the former.

Hence, our final candidates for a model were The FBProphet model and the Exponential Smoothing Forecast. Here on, the selection of the model was on a trial and error implementation basis as we explored the comparison two models by trying it out manually for comparison after our research on what would fit our data, before finalizing the algorithmic procedure our model would follow. So as to what separated these models, in terms of algorithmic process, procedure and methodology were by definition as follows:

- **FbProphet** is A novel time-series forecasting procedure implemented in R and Python by Facebook (implemented in Python by us), that builds on the Additive Procedure Model of Prophet. It is very easy to use (considering similar models like ARIMA which would require a lot of fine tuning for the autoregressive p,d,q variables) and by default Prophet fits additive seasonality – meaning the effect of the seasonality is added to the trend to get the forecast, even if it isn’t present as an attribute. Best fits Linear Data, with scope for expansion

- **Exponential Smoothing Forecast** is An ML procedure amongst the many forecasting procedures used within ArcGIS Pro as part of its space-time pattern mining geoprocessing toolset, that works based on applying weighted average to time series data (working with netCDF file formats). One of the oldest procedures used to support data with a systematic trend and often as an alternative to Box-Jenkins ARIMA.

For the comparison, let's start off with forecasting the temperature values for a few states based on the dataset that we've aggregated for annual temperature.

1. Temperature Forecast using FbProphet

The FBProphet execution in the Google Collab notebook is as follows :

```
import numpy as np
import pandas as pd
from scipy import stats
import statsmodels.api as sm
import matplotlib.pyplot as plt

#Importing the Datasets
df=pd.read_csv('tas_timeseries_annual_cru_1901-2020_IND.csv')
df.head()
```

	Year	India	Andaman and Nicobar	Andhra Pradesh	Assam	Delhi	Goa	Gujarat	Haryana	Himachal Pradesh	...	Bihar	Chandigarh	Chhattisgarh	Dadra and Nagar Haveli	Daman and Diu	Jharkhand	Madhya Pradesh	Puducherry	Uttar Pradesh	Uttarakhand
0	1901	24.26	27.81	27.04	22.77	25.27	26.26	26.66	25.09	9.80	...	25.03	23.56	25.40	25.87	26.08	24.59	25.40	27.78	25.47	13.88
1	1902	24.52	27.84	27.36	22.50	25.52	26.64	27.19	25.34	10.01	...	25.07	23.81	25.72	26.48	26.67	24.66	25.82	27.81	25.67	14.11
2	1903	23.96	27.63	26.77	22.73	24.90	25.96	26.20	24.68	9.40	...	24.91	23.19	25.18	25.52	25.68	24.45	25.03	27.58	25.16	13.58
3	1904	24.02	27.29	26.83	22.51	24.95	26.07	26.59	24.77	9.57	...	24.70	23.28	25.15	25.86	26.07	24.28	25.13	27.37	25.11	13.59
4	1905	23.87	27.80	26.96	22.11	24.65	26.13	26.25	24.43	9.03	...	24.41	22.81	25.10	25.62	25.72	24.06	25.02	27.72	24.90	13.22

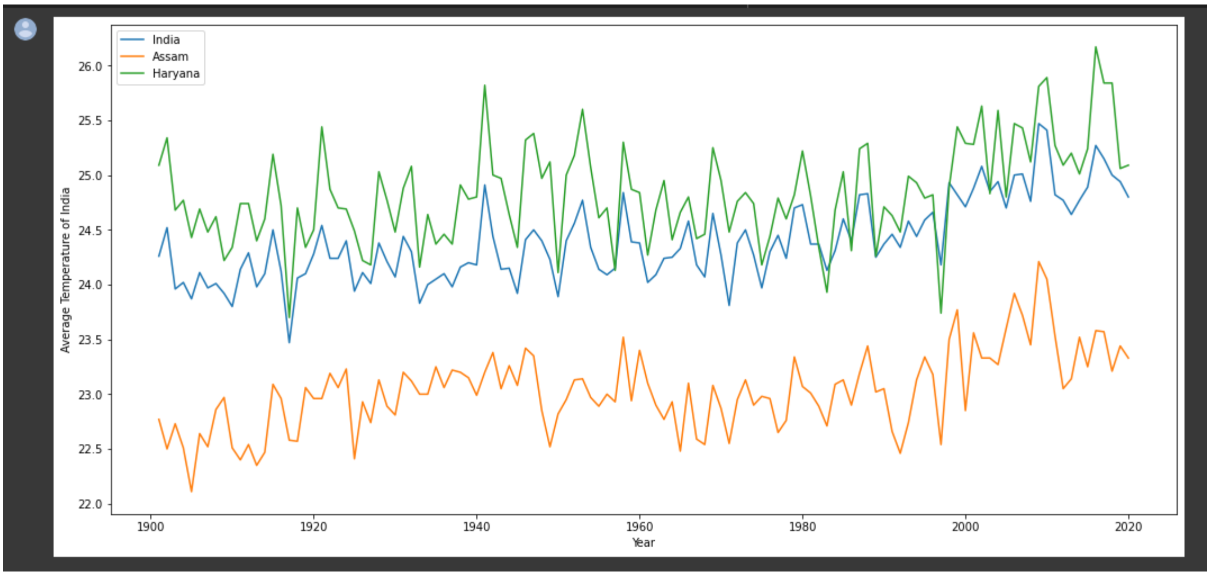
```
#Data cleaning and transformation - Clearing the NA Values and dropping
unnecessary columns, while keeping the date in the required format

#df['Year']=pd.to_datetime(df['Date'])
df.dropna()
#df=df.drop('Time',axis=1)
```

	Year	India	Andaman and Nicobar	Andhra Pradesh	Assam	Delhi	Goa	Gujarat	Haryana	Himachal Pradesh	...	Bihar	Chandigarh	Chhattisgarh	Dadra and Nagar Haveli	Daman and Diu	Jharkhand	Madhya Pradesh	Puducherry	Uttar Pradesh	Uttarakhand
0	1901	24.26	27.81	27.04	22.77	25.27	26.26	26.66	25.09	9.80	...	25.03	23.56	25.40	25.87	26.08	24.59	25.40	27.78	25.47	13.88
1	1902	24.52	27.84	27.36	22.50	25.52	26.64	27.19	25.34	10.01	...	25.07	23.81	25.72	26.48	26.67	24.66	25.82	27.81	25.67	14.11
2	1903	23.96	27.63	26.77	22.73	24.90	25.96	26.20	24.68	9.40	...	24.91	23.19	25.18	25.52	25.68	24.45	25.03	27.58	25.16	13.58
3	1904	24.02	27.29	26.83	22.51	24.95	26.07	26.59	24.77	9.57	...	24.70	23.28	25.15	25.86	26.07	24.28	25.13	27.37	25.11	13.59
4	1905	23.87	27.80	26.96	22.11	24.65	26.13	26.25	24.43	9.03	...	24.41	22.81	25.10	25.62	25.72	24.06	25.02	27.72	24.90	13.22
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115	2016	25.27	27.69	28.23	23.58	26.45	27.25	27.61	26.17	11.18	...	25.81	24.83	26.46	26.85	27.03	25.31	26.29	29.00	26.56	15.26
116	2017	25.15	27.58	28.29	23.57	26.07	27.39	27.56	25.84	10.78	...	25.57	24.45	26.40	26.89	27.04	25.08	26.17	28.87	26.26	14.87
117	2018	25.00	27.32	28.03	23.21	26.03	27.27	27.74	25.84	10.74	...	25.20	24.42	26.07	26.97	27.19	24.65	26.04	28.72	26.12	14.81
118	2019	24.94	27.76	28.37	23.44	25.28	27.39	27.41	25.06	9.95	...	25.37	23.56	26.29	26.79	26.96	24.96	25.94	29.08	25.74	14.05
119	2020	24.80	27.80	28.06	23.33	25.31	27.28	27.44	25.09	10.07	...	25.18	23.71	25.98	26.80	27.04	24.69	25.75	28.93	25.62	14.14

120 rows × 46 columns

```
#Using Matplotlib to plot out comparisons of the temperature changes over
the years between a few states

plt.figure(figsize=(17,8))
plt.plot(df['Year'],df['India'],label='India')
plt.plot(df['Year'],df['Assam'],label='Assam')
plt.plot(df['Year'],df['Haryana'],label='Haryana')
plt.xlabel('Year')
plt.ylabel('Average Temperature of India')
plt.legend()
```

```
plt.show()
```




```
#Importing FBProphet and enabling its modules

!pip install fbprophet
from fbprophet import Prophet
import logging

logging.getLogger().setLevel(logging.ERROR)



#FBProphet considers data to be in the format of the attribute variable y
(Temperature of a state here) and the date variable ds in the yyyy-mm-dd
format for yearly values Hence converting formats

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

	y	ds
0	6	2010-12-01
1	12	2010-12-02
2	48	2010-12-03
3	8	2010-12-05
4	6	2010-12-06

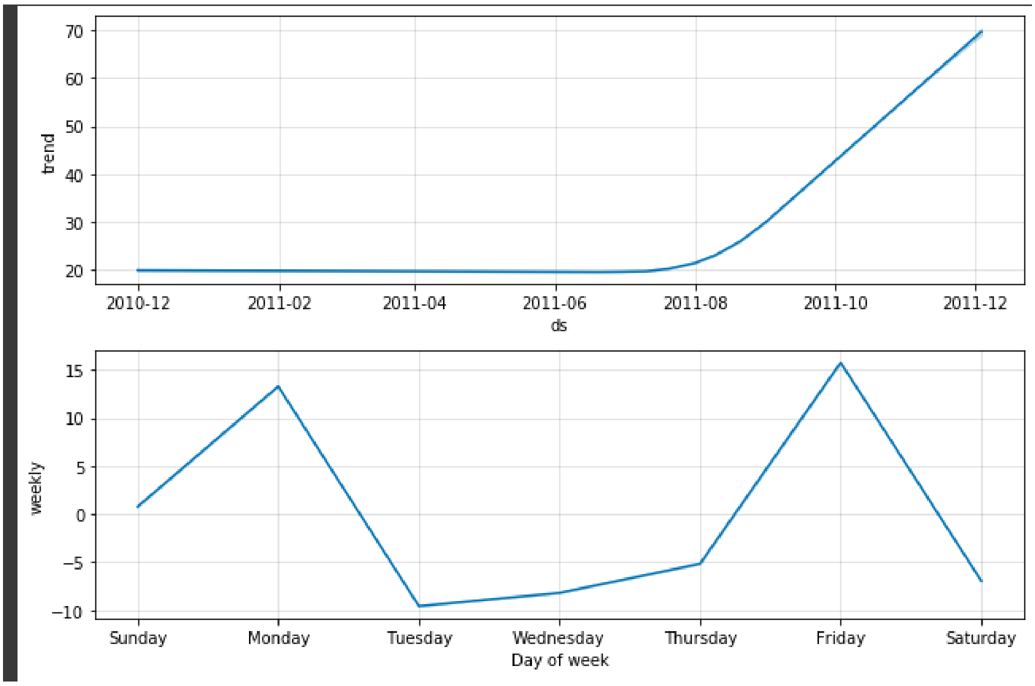
```
#Splitting the datasets into training prediction sizes. We'll be training
on 90% of the data and predicting the remainder 10%
```

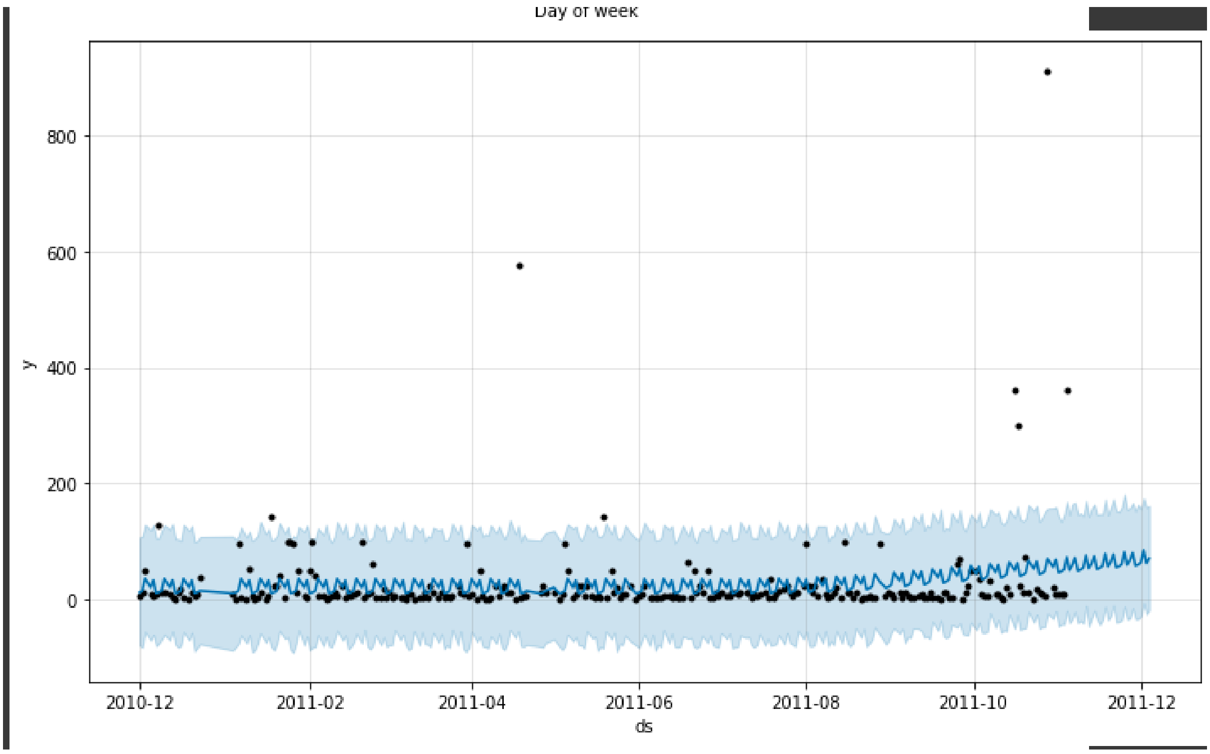
```
prediction_size=int(0.1*len(df))
train_df=df[:-prediction_size]
print(prediction_size)



#Calling the necessary FBProphet Module. Fitting the model within the
variable m. Using the trained model to predict future values in the
forecast variable

m = Prophet()
m.fit(train_df)
future= m.make_future_dataframe(periods=prediction_size)
forecast=m.predict(future)
forecast.head()
```

```
INFO:fbprophet:Disabling yearly seasonality. Run prophet with yearly_seasonality=True to override this.
INFO:fbprophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
```

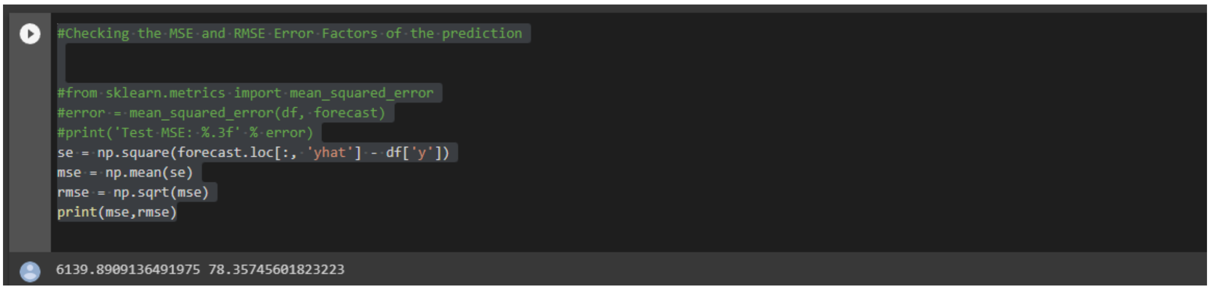
	ds	trend	yhat_lower	yhat_upper	trend_lower	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	weekly	weekly_lower	weekly_upper	multiplicative_terms	multiplicative_terms_lower	multiplicative_t
0	2010-12-01	19.889329	-79.463875	107.006292	19.889329	19.889329	-8.195652	-8.195652	-8.195652	-8.195652	-8.195652	-8.195652	0.0	0.0	
1	2010-12-02	19.887454	-82.999859	108.705877	19.887454	19.887454	-5.165161	-5.165161	-5.165161	-5.165161	-5.165161	-5.165161	0.0	0.0	
2	2010-12-03	19.885579	-56.025833	129.033139	19.885579	19.885579	15.784535	15.784535	15.784535	15.784535	15.784535	15.784535	0.0	0.0	
3	2010-12-05	19.881830	-71.204875	119.289679	19.881830	19.881830	0.782389	0.782389	0.782389	0.782389	0.782389	0.782389	0.0	0.0	
4	2010-12-06	19.879955	-60.986093	128.823756	19.879955	19.879955	13.307755	13.307755	13.307755	13.307755	13.307755	13.307755	0.0	0.0	

```
#Plotting the acquired forecasts. As you can see, FBProphet has its own
inner definitions to find seasonalities that may or may not be defined in
the dataset.

m.plot(forecast)
m.plot_components(forecast)
```






```
#Checking the MSE and RMSE Error Factors of the prediction


#from sklearn.metrics import mean_squared_error
#error = mean_squared_error(df, forecast)
#print('Test MSE: %.3f' % error)
se = np.square(forecast.loc[:, 'yhat'] - df['y'])
mse = np.mean(se)
rmse = np.sqrt(mse)
print(mse,rmse)
```



```
#Checking the MSE and RMSE Error Factors of the prediction


#from sklearn.metrics import mean_squared_error
#error = mean_squared_error(df, forecast)
#print('Test MSE: %.3f' % error)
se = np.square(forecast.loc[:, 'yhat'] - df['y'])
mse = np.mean(se)
rmse = np.sqrt(mse)
print(mse,rmse)

6139.8909136491975 78.35745601823223
```

One major advantage that could be seen of the FBProphet model is that, despite the very few attributes present in the data, the algorithm works in such a way so as to show **varying seasonality factors** (by months, days and weeks), even if it does not exist as a dimension in the dataset. It hence gives way more insights than a comparable model with much less efforts.

But from the outputs, its also to be noted that, considering that we had about 19 years of data with around 12,000 rows, the amount of inconsistency in the data could be massive. From the final forecast plot, it is very visible that the **outlier values** have been way off the range, some even going above 400. It is visible that the model is not handling such outliers in a not-so-linear data, as well as it should. Hence as we can see, the regression error values in MSE, RMSE have been pretty high. The graph also shows the growing deviation of the model. So even though, the model gives a formidable result, it is not reliable when it comes to data such as ours.

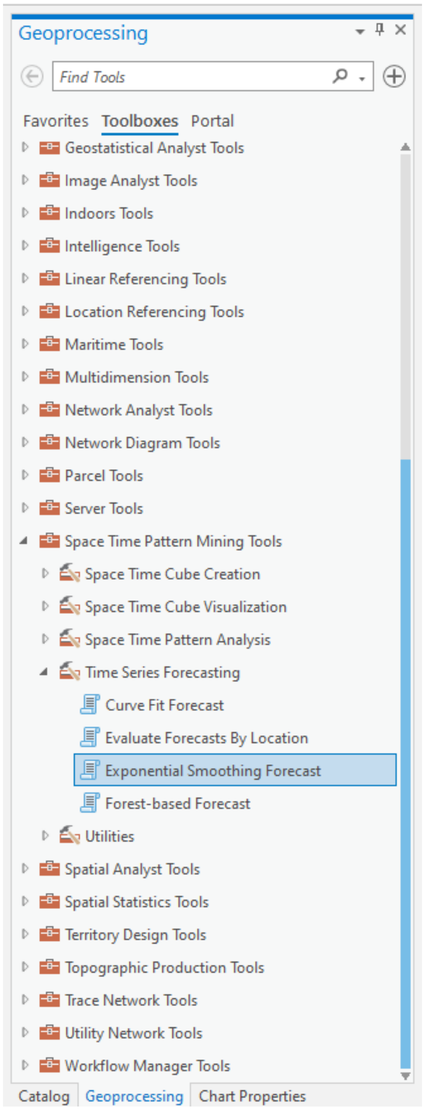
This is exactly where the Exponential Smoothening tool of ArcGIS came in Handy.

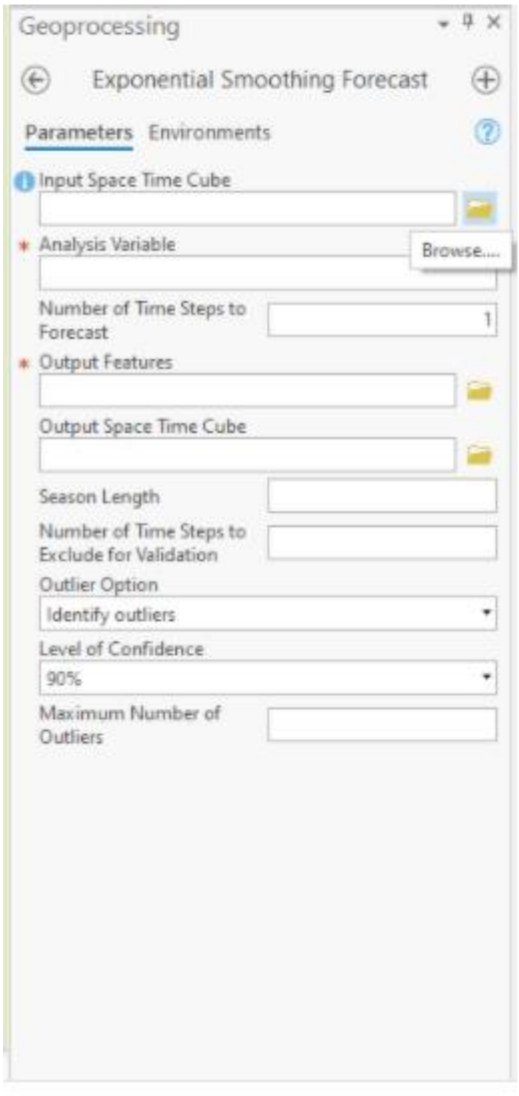
2. Temperature Forecast using Exponential Smoothening

Exponential Smoothening was among the 4 models present for time series forecasting in ArcGIS under the Space-Time Pattern Mining Tools.

Unlike FBProphet, which could be used for multivariate data at times, this is a univariate exclusive model. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older.

This necessarily means that it makes it a good model for short-mid term forecasts especially with respect to recent history of non linear data, because as the weights decay over time, more importance would be given to accommodating recent trends in the dataset, despite how large, and the prediction could be more in par with recent changes. For example in the current dataset of values over a 100 years, values from probably 2010-2020 would be given more accurate preference and less bias as compared to data from 1950-1960.





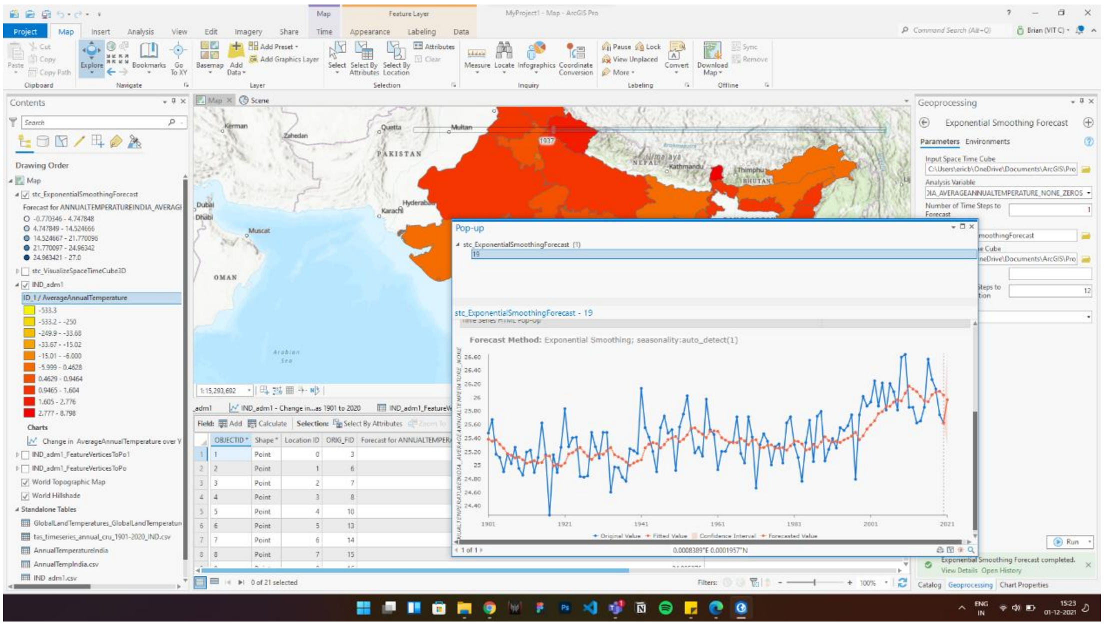
Forecasting within ArcGIS is a very straightforward process and unlike FBProphet, requires no programming knowledge at all. Though, if ultimately necessary for advanced fine-tuning, ArcGIS does provide the ability for scripting and using languages like Python for extensive capabilities. But again, this is purely upto choice, and as a GIS software, it does exactly that, taking the heavy memory load of the user.

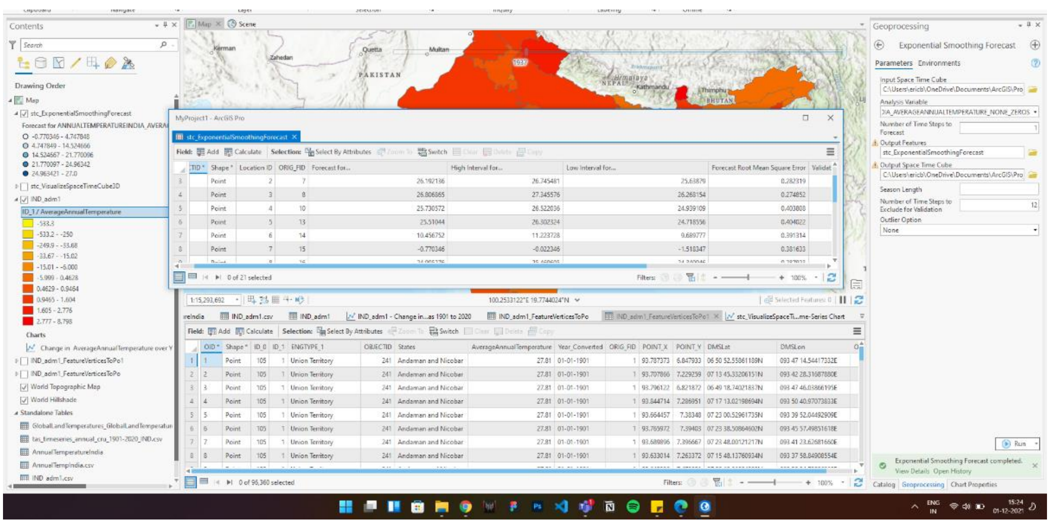
Here again is yet another viable example application of why conversion to a format such as NetCDF is so flexible. Since we're working with the computation of Temporal Data, ArcGIS expects the input format to be a multidimensional vector. Since, we've already created that for the Space Time Cube, we could use the same file as the input for the Exponential Smoothening Forecast tool

Now one of the major improvements, when it comes to this over FbProphet is the way it learns from data, but moreover the way it handles the data. Here, as visible from the processing tab, it provides us with an option to let it handle outliers on its own, based on a confidence factor that its algorithm applies for the data.

This means that values like 400 in temperature, which gave rise to such a high fluctuation in our prediction with FbProphet, will be effectively managed within ArcGIS itself, while handling other inconsistencies in the data.

Once the assigning of attributes is done, the software takes its time for the computation without requiring any fine tuning from the user, with it ending up creating another **output space time cube with additional forecast-based features** and attributes (such as higher trend, lower trend, range of change, forecasted temperature, etc.). The results are immediately represented in the map and as a graph as well.

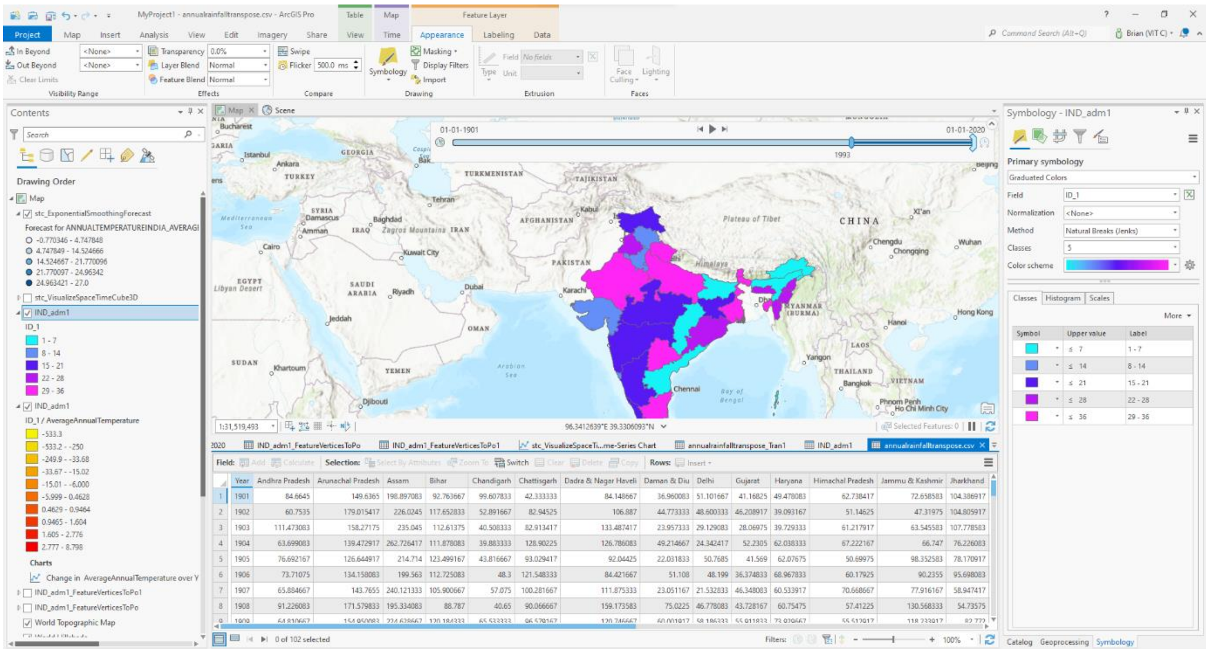




Another huge advantage of this function is also that, while FBProphet had to predict seasonality for each state separately, the Exponential Smoothing function predicts future forecasts for a short-term period ahead **for every single attribute at once**, taking around the same time, cementing the fact on how powerful the model is.

And the cherry on top is without doubt, the accuracy. With zero to little fine tuning as mentioned, the software handles outliers and inconsistencies for over 12,000 data points with ease and predicts a very considerable forecast for a year ahead based on its learnings.

As for the insights, the attribute table of the prediction shows trend values, ranges of higher/lower end, predicted temperature values, etc. With the new space time cube output of the forecast over the layer, clicking on each state, would provide you of a popup not just showing the attribute values, but also a Graph of the forecast as well as the trend variation over the years of that particular location, as seen in the image. The blue lines are the variations, while the red line denotes the learning of the trend. There is a dotted point which signifies the point to which we do have existing data and the the red line crossing it symbolizes the forecast curve for the temperature of that particular location. The exponential smoothing has averaged a prediction of **0.79 Degree Celsius increase in the temperature globally by 2021** when taking values all over India for 2021 from the data learned from 1901 till 2020.



The exact same process was carried out in parallel for the precipitation datasets as well. Hence the Time-Series forecast of two attributes, Temperature and

Precipitation was carried out independently.for each state as well as the national average was also computed after a trial and error comparison of multiple models to find out the one that fit our model the most.

RESULT

As per our target goal, we've succeeded in generating a case study visually to represent the variation of attributes in regards to climate change to document the variations. We've also set up a Time-Series Forecasting model within ArcGIS- - our preferred GIS Software, based on the data we analyzed which provided us insights on varying temperature trends across India and its near future in that regard. The biggest success is however, the confirmation that our model is on the right path to analyze and predict possible future scenarios if fed the right amount of data.

As per a news article from Hindustan Times we collated during our research phase, there was a prediction by the ministry of Earth Sciences that the temperature rise between 1901 and 2018 in India has been around **0.7 degree Celsius**, and forecasted 2021 to be between **0.7 and 0.**8 while the rise at the end of the 21st century could be by a whopping 4%.


Home / India News / Temperature over India likely to rise by over 4 degrees Celsi...

INDIA NEWS

Temperature over India likely to rise by over 4 degrees Celsius by end of 21st century: Govt report

India's average temperature has risen by around 0.7 degree Celsius during 1901–2018, largely on account of Green House Gases-induced warming, states the report by the Ministry of Earth Sciences.

The report has been prepared by the Centre for Climate Change Research, a cell under MoES' Indian Institute of Tropical Meteorology, Pune.(AFP)

Updated on Jun 15, 2020 03:45 PM IST

New Delhi |By Press Trust of India| Posted by Susmita Pakrasi

India

Assessment of climate change over the Indian region: A report of the Ministry of Earth Sciences (MoES), Government of India

Assessment • **Source:** Govt. India • **Posted:** 17 Jun 2020 • **Originally published:** 17 Jun 2020 • **Origin:** View original

Executive Summary

Observed Changes in Global Climate

The global average temperature has risen by around 1°C since pre-industrial times. This magnitude and rate of warming cannot be explained by natural variations alone and must necessarily take into account changes due to human activities. Emissions of greenhouse gases (GHGs), aerosols and changes in land use and land cover (LULC) during the industrial period have substantially altered the atmospheric composition, and consequently the planetary energy balance, and are thus primarily responsible for the present-day climate change. Warming since the 1950s has already contributed to a significant increase in weather and climate extremes globally (e.g., heat waves, droughts, heavy precipitation, and severe cyclones), changes in precipitation and wind patterns (including shifts in the global monsoon systems), warming and acidification of the global oceans, melting of sea ice and glaciers, rising sea levels, and changes in marine and terrestrial ecosystems.

Projected Changes in Global Climate

Global climate models project a continuation of human-induced climate change during the twenty-first century and beyond. If the current GHG emission rates are sustained, the global average temperature is

Download document

(PDF | 23.95 MB)

Primary country: India

Source: Government of India

Disasters: India: Floods - Jun 2012 / India: Floods - Jun 2016 / India: Floods and Landslides - Jun 2015 / India: Floods and Landslides - Jun 2019 / India: Floods and Landslides - May 2017 / India: Floods and Landslides - Sep 2014 / India/Nepal: Floods and Landslides - Jun 2013

Format: Assessment

Themes: Agriculture / Climate Change and

Now this was a point of reference for us, and since our forecast showed a rise around the range of **0.6 -0.9 Degree Celsius across states**, we could conclude that the prediction is in par with expected results based on trends. Hence, this developed model is reliable enough to be worked on for future long term forecasts.

The entire ArcGIS project alongside the utilized datasets, shapefiles, Google Collab notebooks and additional assets are available for preview on the Project GitHub link at :

https://github.com/EricBrianAnil/GISProj

FUTURE SCOPE

As for what lies beyond the J Component for this project, our primary motives would be as follows :

- **Granulate the values for smaller territories :** Scrutinize for smaller boundaries such as district boundaries to predict a more accurate fluctuation of the attributes. This would require efforts in proper data collection and manipulation to create a reliable source of consistent data to the maximum extend so as to be used by us or others in the future.

- **Scaling the map :** Our initial target, to globalize the analysis from India to view effects on a larger scale, especially countries that project more fluctuation in their Data
- **Univariate to Multivariate Data :** When considering trends in smaller parts of the states towards climate change, the number of attributes could be bigger, making it a multivariate model for execution. The Starting point could be to collate a list of dependant attribute and create a score system based on impact to predict climate change based on these factors.

- **Open Sourcing the Project for for better data / prediction quality and insights :** Our end product of this project is to make it a multidimensional vector NetCDF file consisting of current forecasts that could help people use the map to improve, add on data and make valuable predictions using the model we've applied to the data we've arranged, since we've confirmed its reliability based on studies. Hence our end product would be for a geo document / map that enables people to forecast local climate change trends and modify as per requirement.

CONCLUSION

The first half of the J component project has primarily involved exploration of the arcGIS Pro Desktop GIS Toolkit by ESRI using documentation and tutorial references provided by ESRI itself. We've collected the necessary data to work out analysis on the initial phase of our project - Data regarding temperature variations and precipitation across different states in India as our primary domain. The time variant data has been processed and mapped for visualisation into an state-based shapefile map of India using arcGIS pro and the temperature variations across the regions has been classified via the symbology feature.

Furthermore, for the remainder of the project, our target had been set towards analysing these data points to predict a forecast of these variations for the nearby future alongside accommodating a few more relevant attributes. Primarily by helping us mould a decent time-series model stitched for the purpose of tracking climate change along the way. These would be visualised and charted with focus on providing better accuracy catering to India and its localised regions, making an apt model for future predictions and with room for flexibility for data collection and improvement as an open sourced GIS Project.

REFERENCES & RESOURCES

1. ESRIs analysis of factors affecting climate change globally (https://www.esri.com/about/newsroom/publications/wherenext/nexttech/climate-change/)

2. ESRI Article on Climate Science based Initiatives (https://www.esri.com/en-us/about/science/initiatives/weather-climate-science/)

3. An ESRI Overview about fighting the climate crisis together from experts (https://www.esri.com/en-us/about/climate-change/overview)

4. Climate change Assessment over Indian Region – A Study by the ministry of Earth Sciences (MOES) (https://reliefweb.int/report/india/assessment-climate-change-over-indian-region-report-ministry-earth-sciences-moes)

5. A Hindustan Times news article regarding the inflation in Temperature variations over India (https://www.hindustantimes.com/india-news/temperature-over-india-likely-to-rise-by-over-4-degress-celsius-by-end-of-21st-century-govt-report/story-ZsXcDjcYv7DiHWG7F8egGO.html)

6. Climate Data Repo by NCAR's GIS Program (https://gisclimatechange.ucar.edu)

7. Climate Data Service Portal of the Indian Metereological Department (https://cdsp.imdpune.gov.in)

8. NOAA Central Dataset Library (https://library.noaa.gov)

9. Climate Change Knowledge Portal maintained by the World Bank Group (https://climateknowledgeportal.worldbank.org)

10. DIVA GIS Resources, primarily spatial (https://www.diva-gis.org/)

11. Data World Repository for Datasets (https://data.world/data-society/global-climate-change-data) (https://data.world/rajanand/rainfall-in-india)

12. Climate Change as a Geographical Problem : A publication by ESRI
(https://www.esri.com/library/ebooks/climate-change.pdfTable/)

13. A Spatial Data Webinar on Evaluating Climate Change Data
(https://www.youtube.com/watch?v=7GTtCtXJJ0Y)

14. FBProphet Documentation by Facebook
(https://facebook.github.io/prophet/docs/installation.html#python)

15. Spatial Time Patter Mining Tool Reference in ArcGIS Pro
(https://pro.arcgis.com/en/pro-app/latest/tool-reference/space-time-pattern-mining/exponentialsmoothingforecast.htm)