

Deccan Education Society’s

NAVINCHANDRA MEHTA INSTITUTE OF TECHNOLOGY AND DEVELOPMENT

**NAAC Accredited “B++”**

**WEATHER PREDICTION SYSTEM**

SUBMITTED BY

**Pratik Siriah  
C22118**

Mentor

**Mr. Ganesh Bhagwat**

**2022 - 2023**

 Submitted to University of Mumbai

in partial fulfillment of the requirements for qualifying

MASTER OF COMPUTER APPLICATION   
 Examination

**Deccan Education Society’s**

NAVINCHANDRA MEHTA INSTITUTE OF TECHNOLOGY AND DEVELOPMENT

PROJECT CERTIFICATE

This is to certify that the Project done at **DES NMITD** by **Mr. Pratik Siriah (**Seat No. **C22118)** in partial fulfillment for MCA Degree Examination has been found satisfactory. This report had not been submitted for any other examination and does not form part of any other course undergone by the candidate.

Internal Guide Director

EXAMINED BY

EXTERNAL EXAMINER ………………………………

DATE:

College Stamp

**Acknowledgement**

I would like to express my profound gratitude to Dr. Anita Bobade, Director of DES NMITD and Dr. Rasika Mallya, HOD of MCA department, and for their contributions to the completion of my project titled Hospital Management System.

Thanks to all our teachers Dr. Sulakshana Vispute, Mrs. Apeksha Gaje, Mrs. Swapnali Mahadik, Mr. Pratik Desai, Mr. Pritam Warke, for helping and guiding me throughout my project.

I would like to express my special thanks to our mentor Mr. Ganesh Bhagwat for his time and efforts he provided throughout the year. Your useful advice and suggestions were really helpful to me during the project’s completion. In this aspect, I am eternally grateful to you.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Index Topic** | **Page** |
| 1 | **Acknowledgement** | **3** |
| 2 | **Introduction** | **5** |
|  | **Literature Survey** | **7** |
| 3 | **Problem statement** | **8** |
| 4 | **Gnatt chart** | **9** |
| 6 | **System Design** | **10** |
|  | **System Architecture** | **11** |
| 7 | **Software Requirements** | **14** |
| 8 | **Feasibility Study** | **18** |
| 9 | **System Implementation** | **20** |
| 10 | **Screen Layouts & Report Layouts** | **23** |
| 11 | **Conclusion** | **31** |
| 12 | **Future Enhancement** | **31** |
| 13 | **References** | **32** |

1. **Introduction**

Weather forecasting is the application of science and technology to predict the state of the atmosphere for a given location. Ancient weather forecasting methods usually relied on observed patterns of events, also termed pattern recognition. For example, it might be observed that if the sunset was particularly red, the following day often brought fair weather. However, not all of these predictions prove reliable. In this work, I have made an effort to predict the weather by applying Machine Learning techniques and taking different parameters such as temperature, humidity, and wind.

Once calculated manually based mainly upon changes in barometric pressure, current weather conditions, and sky condition or cloud cover, weather forecasting now relies on computer-based models that take many atmospheric factors into account. Human input is still required to pick the best possible forecast model to base the forecast upon, which involves pattern recognition skills, teleconnections, knowledge of model performance, and knowledge of model biases.

A weather forecasting system takes parameters and will forecast weather based on previous records therefore this prediction will prove reliable. This system can be used in Air Traffic, Marine, Agriculture, Forestry, Military, Navy, etc. Forecasting the weather on a particular day and date is the main aim of this project. In this project, I used several data resources in form of datasets. National Centers for environmental information (NOAA), Kaggle, and several other similar platforms provided data that is used in weather predictions. I obtained data from these platforms for weather predictions and after applying ML algorithms the weather is predicted and displayed.

**Introduction to Machine Learning:**

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. It gives the computer that makes it more similar to humans i.e. ability to learn. Machine learning is used in many streams than anyone would accept. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Machine Learning is a sub-area of artificial intelligence, whereby the term refers to the ability of IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to recognize patterns on the basis of existing algorithms and data sets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is generated on the basis of experience. In order to enable the software to independently generate solutions, the prior action of people is necessary. For example, the required algorithms and data must be fed into the systems in advance and the respective analysis rules for the recognition of patterns in the data stock must be defined. Once these two steps have been completed, the system can perform the following tasks by Machine Learning:

• Finding, extracting and summarizing relevant data

• Making predictions based on the analysis data

• Calculating probabilities for specific results

Basically, algorithms play an important role in Machine Learning: On the one hand, they are responsible for recognizing patterns and on the other hand, they can generate solutions.

1. **LITERATURE SURVEY**

Machine learning in weather forecasting is a recent trend in the literature. There are several works that discuss this topic. Holmstrom et al. proposed a technique to forecast the maximum and minimum temperature of the next seven days, given the data of the past two days. They utilized a linear regression model, as well as a variation of a functional linear regression model. They showed that both the models were outperformed by professional weather forecasting services for the prediction of up to seven days. However, their model performs better in forecasting later days or longer time scales. A hybrid model that used neural networks to model the physics behind weather forecasting was proposed by Krasnopolsky and Rabinowitz. Support vector machines were utilized for weather prediction as a classification problem by Radhikaet al.

A data mining-based predictive model to identify the fluctuating patterns of weather conditions was proposed. The patterns from historical data are used to approximate the upcoming weather conditions. The proposed data model uses Hidden Markov Model for prediction and k-means clustering for extracting weather condition observations. Grover et al. studied weather prediction via a hybrid approach, which combines discriminatively trained predictive models with deep neural networks that model the joint statistics of a set of weather-related variables. Montori et al. used the concept of crowdsensing, where participating users share their smartphone data about environmental phenomena. They introduced an architecture named SenSquare, which handles data from IoT sources and crowdsensing platforms and displays the data unified to subscribers. This data is used in smart city environment monitoring. However, none of these works use the idea of combining data from neighboring places.

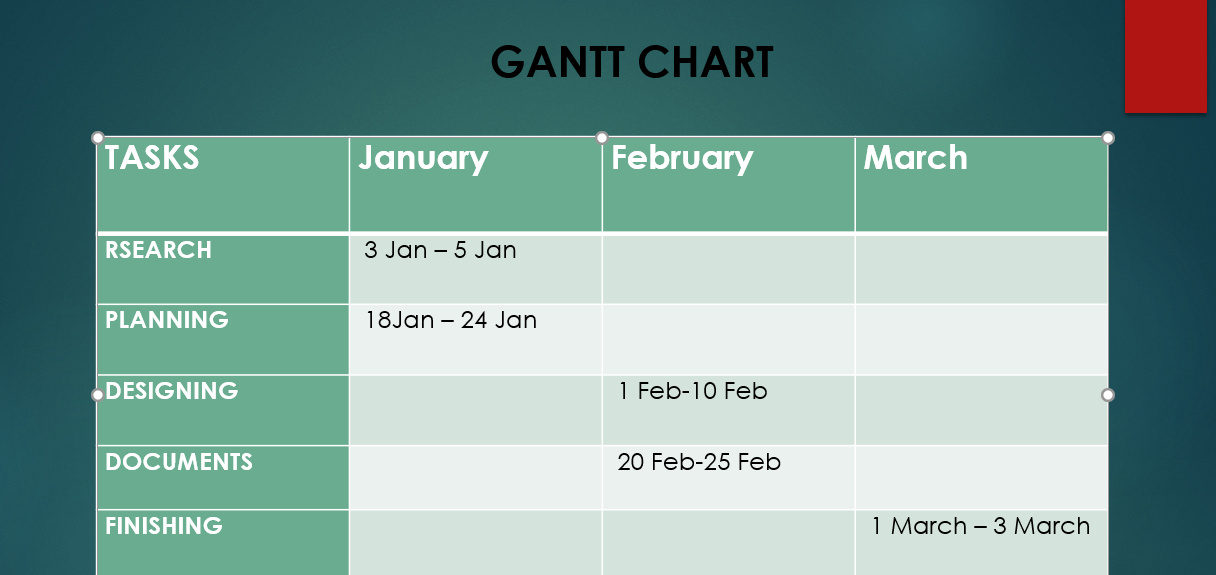
1. **Problem Statement**

My Project concentrates on predicting the weather using Machine learning methods and taking different weather parameters. In an attempt to predict stock market trends and future stock prices, market researchers, investors, and scholars regularly propose a range of models. These models are based on various methods including the following.

• Single and dual sources of information. Single information source methods either utilize numerical or semantic information extracted from news and reviews while dual-source methods utilize historical price and volume data as well.

I am performing the dual-source of an information model for my topic which predicts the outcome on the basis of historical price and volume data and is using machine learning algorithms.

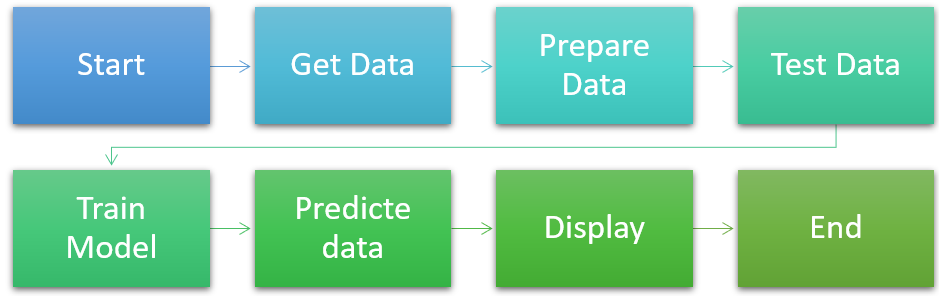
1. **Gnatt Chart**

****

1. **System Design**

Data Flow Diagram :

* A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.
* Notations :
* Process
* Dataflow

****

1. **System Architecture**

System Architecture describes “the overall structure of the system and the ways in which the structure provides conceptual integrity”. The system architecture to build a recommendation system involves the following five major steps.

1. Data Acquisition

2. Data Pre-processing

3. Feature Extraction

4. Training Methods

5. Testing Data

1. **Data Acquisition:** The goal of this step is to find and acquire all the related datasets or data sources. In this step, the main aim is to identify various available data sources, as data are often collected from various online sources like databases and files. The size and the quality of the data in the collected dataset will determine the efficiency of the model. The weather dataset is collected from the kaggle website

**2. Data Pre-processing**: The goal of this step is to study and understand the nature of data that was acquired in the previous step and also to know the quality of data. In this step, we will check for any null values and remove them as they may affect the efficiency. Identifying duplicates in the dataset and removing them is also done in this step.

**3. Feature Extraction**: Usually, dataset will be split into train and test in the ratio of 8:2 i.e., 80 percent of data is used for training and 20 percent of data is used for testing the model. We have also done in the same way.

**4. Training Methods:**  Now, we have our training and testing data. The next step is to identify the possible training methods and train our models. We have used KMeans clustering method for training models.

* Kmeans:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters in such a manner that each dataset belongs to only one group that has similar features. Here K defines the number of pre-defined clusters. We have to associate each cluster with a centroid in this algorithm. The sum of distances between the data point and their corresponding clusters should be minimized. The unlabeled dataset is taken as an input and the dataset into k-number of clusters is divided, and the process is repeated until it does not find the best clusters. We have to predetermine the k value in this algorithm. Elbow method is used to find the value of k which decides the number of clusters. This method uses the Within Cluster Sum of Squares (WCSS) value that defines the total variations within a cluster.

The Formula for calculating the value of WCSS for n clusters is as follows:

WCSS = ∑Pi in Cluster1distance (Pi C1)2 +∑Pi in Cluster2distance (Pi C2)2+…..+∑Pi in Cluster n distance(Pi Cn)2

The basic steps involved in K-Means Clustering algorithm is as follows:

**Step-1:** Select the number K which gives the number of clusters.

**Step-2:** Select random K number of points or centroids.

**Step-3:** Each data point to their nearest centroid should be assigned, which forms the predefined K clusters.

**Step-4:** Calculate the variance and place a new centroid for each cluster.

**Step-5:** We have to repeat the step-3, each data-point to the new closest centroid of each cluster should be reassigned.

**Step-6:** If reassignment happens, then go to step-4 or else go to step7.

**Step-7:** Stop.

**System Requirements**

A requirement is a feature that the system must have or a constraint that it must to be accepted by the client. Requirement Engineering aims at defining the requirements of the system under construction. Requirement Engineering include two main activities requirement elicitation which results in the specification of the system that the client understands and analysis which in analysis model that the developer can unambiguously interpret. A requirement is a statement about what the proposed system will do.

Requirements can be divided into two major categories:

• Functional Requirements.

• Non-Functional Requirements.

**Functional Requirements:**

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements describe the interactions between the system and its environment independent of its application.

• Applying the algorithms on the train data

• Display the recommendations by the model.

**Non-Functional Requirements:**

Non-Functional Requirements specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs.

• Accuracy

• Reliability

• Flexibility

1. **Software Requirements**

**Purposed Method**:

* Decision Tree Regression:
  + Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.
  + Discrete output example: A weather prediction model that predicts whether or not there’ll be rain on a particular day.
  + Continuous output example: A profit prediction model that states the probable profit that can be generated from the sale of a product.

**Software Requirement:**

* + Python 3.6.x
  + Environment:
    - Visual Studio Code
    - Jupyter Notebook
  + Framework:
    - Sklearn

**Hardware Requirement:**

* + Modern Operating System:
    - Windows 7 or 10
    - Mac OS X 10.11 or higher, 64-bit
    - Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)
  + x86 64-bit CPU (Intel / AMD architecture)
  + 8 GB RAM
  + 5 GB free disk space

1. Python:
   1. Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.
   2. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Features :

* Easy Language
* Python is an easy language. It is easy to read, write, learn and understand.
* Python has a smooth learning curve. It is easy to learn.
* Python has a simple syntax and Python code is easy to understand.
* Since it’s easy to understand, you can easily read and understand someone else’s code.
* Python is also easy to write because of its simple syntax.
* Because it is an easy language, it is used in schools and universities to introduce students to programming. Python is for both startups and big companies.
* The Python language is designed to make developers life easy. Reading a Python code is like reading an English sentence. This is one of the key reason that makes Python best for beginners.
* Python uses indentation instead of curly braces, unlike other programming languages. This makes the code look clean and easier to understand.
* Python is an interpreted language. It comes with the IDLE (Interactive Development Environment). This is an interpreter and follows the REPL structure (Read-Evaluate-Print-Loop).

1. Jupyter Notebook:
   1. The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process: developing, documenting, and executing code, as well as communicating the results. The Jupyter notebook combines two components
   2. A web application: a browser-based tool for interactive authoring of documents which combine explanatory text, mathematics, computations, and their rich media output.
   3. Because you use Jupyter in a web browser, some people are understandably concerned about using it with sensitive data. However, if you followed the standard install instructions, Jupyter is running on your own computer. If the URL in the address bar starts with http://localhost: or http://127.0.0.1:, it’s your computer acting as the server. Jupyter doesn’t send your data anywhere else and as it’s open source, other people can check that we’re being honest about this.
   4. You can also use Jupyter remotely: your company or university might run the server for you, for instance. If you want to work with sensitive data in those cases, talk to your IT or data protection staff about it.

Main Features of Jupyter Notebook:

* Notebook documents contains the inputs and outputs of a interactive session as well as additional text that accompanies the code but is not meant for execution. In this way, notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics, and rich representations of resulting objects. These documents are internally JSON files and are saved with the .ipynb extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues.
* Notebooks may be exported to a range of static formats, including HTML (for example, for blog posts), restructured Text, LaTeX, PDF, and slide shows, via the nbconvert command.
* Furthermore, any .ipynb notebook document available from a public URL can be shared via the Jupyter Notebook Viewer <nbviewer>. This service loads the notebook document from the URL and renders it as a static web page. The results may thus be shared with a colleague, or as a public blog post, without other users needing to install the Jupyter notebook themselves. In effect, nbviewer is simply nbconvert as a web service, so you can do your own static conversions with nbconvert, without relying on nbviewer.

1. Sklearn (scikit-learn):
   1. Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Features of Sklearn:

* Simple and efficient tools for predictive data analysis
* Accessible to everybody, and reusable in various contexts
* Built on NumPy, SciPy, and matplotlib
* Open source, commercially usable - BSD license

1. Matplotlib:
   1. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Features of Matplotlib:

* Semantic way to generate complex, subplot grids.
* Colored labels in legends
* Ticks and labels: new alignments of labels for axes and color bars

1. Pandas:
   1. Pandas is an open source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named Numpy, which provides support for multi-dimensional arrays. As one of the most popular data wrangling packages, Pandas works well with many other data science modules inside the Python ecosystem, and is typically included in every Python distribution, from those that come with your operating system to commercial vendor distributions like ActiveState’s ActivePython.

Features of Pandas :

* Data cleansing
* Data fill
* Data normalization
* Merges and joins
* Data visualization
* Statistical analysis
* Data inspection
* Loading and saving data

1. Decision Tree Regression:
   1. Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values
2. RNN(Recurrent Neural Network):
   1. RNNs are a powerful and robust type of neural network and belong to the most promising algorithms in use because it is the only one with an internal memory.
   2. Like many other deep learning algorithms, recurrent neural networks are relatively old. They were initially created in the 1980’s, but only in recent years have we seen their true potential. An increase in computational power along with the massive amounts of data that we now have to work with, and the invention of long short-term memory (LSTM) in the 1990s, has really brought RNNs to the foreground.
   3. Because of their internal memory, RNN’s can remember important things about the input they received, which allows them to be very precise in predicting what’s coming next. This is why they're the preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more. Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms.
3. **Feasibility Study**

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

• Economic Feasibility

• Technical Feasibility

• Operational Feasibility

**Economic Feasibility:**

As system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies, there is nominal expenditure and economic feasibility for certain.

**Technical Feasibility:**

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves evaluation of the hardware, software, and other technology requirements of the proposed system. This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

• A brief description of the business to assess more possible factors which could affect the study

• The part of the business being examined

• The human and economic factor • The possible solutions to the problem At this level, the concern is whether the proposal is both technically and legally feasible (assuming moderate cost). The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

**Operational Feasibility:**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation.

Some of the important issues raised are to test the operational feasibility of a project includes the following:

• Is there sufficient support for the management from the users?

• Will the system be used and work properly if it is being developed and implemented?

• Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

1. **System Implementation**

import numpy as np

import pandas as pd

import plotly.express as px

import plotly.graph\_objects as go

from plotly.subplots import make\_subplots

from datetime import datetime

df = pd.read\_csv(r'D:\Project\Weather\_new\_2018.csv',)

df.head()

df1 = pd.melt(df, id\_vars='YEAR', value\_vars=df.columns[1:])

df1.head()

df1['Date'] = df1['variable'] + ' ' + df1['YEAR'].astype(str)

df1.loc[:,'Date'] = df1['Date'].apply(lambda x : datetime.strptime(x, '%b %Y'))

df1.head()

df1.columns=['Year', 'Month', 'Temprature', 'Date']

df1.sort\_values(by='Date', inplace=True) ## To get the time series right.

fig = go.Figure(layout = go.Layout(yaxis=dict(range=[0, df1['Temprature'].max()+1])))

fig.add\_trace(go.Scatter(x=df1['Date'], y=df1['Temprature']), )

fig.update\_layout(title='Temprature Throught Timeline:',

                 xaxis\_title='Time', yaxis\_title='Temprature in Degrees')

fig.update\_layout(xaxis=go.layout.XAxis(

    rangeselector=dict(

        buttons=list([dict(label="Whole View", step="all"),

                      dict(count=1,label="One Year View",step="year",stepmode="todate")

                     ])),

        rangeslider=dict(visible=True),type="date")

)

fig.show()

fig = px.box(df1, 'Month', 'Temprature')

fig.update\_layout(title='Warmest, Coldest and Median Monthly Tempratue.')

fig.show()

from sklearn.cluster import KMeans

sse = []

target = df1['Temprature'].to\_numpy().reshape(-1,1)

num\_clusters = list(range(1, 10))

for k in num\_clusters:

    km = KMeans(n\_clusters=k)

    km.fit(target)

    sse.append(km.inertia\_)

fig = go.Figure(data=[

    go.Scatter(x = num\_clusters, y=sse, mode='lines'),

    go.Scatter(x = num\_clusters, y=sse, mode='markers')

])

fig.update\_layout(title="Evaluation on number of clusters:",

                 xaxis\_title = "Number of Clusters:",

                 yaxis\_title = "Sum of Squared Distance",

                 showlegend=False)

fig.show()

km = KMeans(3)

km.fit(df1['Temprature'].to\_numpy().reshape(-1,1))

df1.loc[:,'Temp Labels'] = km.labels\_

fig = px.scatter(df1, 'Date', 'Temprature', color='Temp Labels')

fig.update\_layout(title = "Temprature clusters.",

                 xaxis\_title="Date", yaxis\_title="Temprature")

fig.show()

fig = px.histogram(x=df1['Temprature'], nbins=200, histnorm='density')

fig.update\_layout(title='Frequency chart of temprature readings:',

                 xaxis\_title='Temprature', yaxis\_title='Count')

df['Yearly Mean'] = df.iloc[:,1:].mean(axis=1) ## Axis 1 for row wise and axis 0 for columns.

fig = go.Figure(data=[

    go.Scatter(name='Yearly Tempratures' , x=df['YEAR'], y=df['Yearly Mean'], mode='lines'),

    go.Scatter(name='Yearly Tempratures' , x=df['YEAR'], y=df['Yearly Mean'], mode='markers')

])

fig.update\_layout(title='Yearly Mean Temprature :',

                 xaxis\_title='Time', yaxis\_title='Temprature in Degrees')

fig.show()

fig = px.line(df1, 'Year', 'Temprature', facet\_col='Month', facet\_col\_wrap=4)

fig.update\_layout(title='Monthly temprature throught history:')

fig.show()

df['Winter'] = df[['DEC', 'JAN', 'FEB']].mean(axis=1)

df['Summer'] = df[['MAR', 'APR', 'MAY']].mean(axis=1)

df['Monsoon'] = df[['JUN', 'JUL', 'AUG', 'SEP']].mean(axis=1)

df['Autumn'] = df[['OCT', 'NOV']].mean(axis=1)

seasonal\_df = df[['YEAR', 'Winter', 'Summer', 'Monsoon', 'Autumn']]

seasonal\_df = pd.melt(seasonal\_df, id\_vars='YEAR', value\_vars=seasonal\_df.columns[1:])

seasonal\_df.columns=['Year', 'Season', 'Temprature']

fig = px.scatter(seasonal\_df, 'Year', 'Temprature', facet\_col='Season', facet\_col\_wrap=2, trendline='ols')

fig.update\_layout(title='Seasonal mean tempratures throught years:')

fig.show()

px.scatter(df1, 'Month', 'Temprature', size='Temprature', animation\_frame='Year')

from sklearn.tree import DecisionTreeRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import r2\_score

df2 = df1[['Year', 'Month', 'Temprature']].copy()

df2 = pd.get\_dummies(df2)

y = df2[['Temprature']]

x = df2.drop(columns='Temprature')

dtr = DecisionTreeRegressor()

train\_x, test\_x, train\_y, test\_y = train\_test\_split(x,y,test\_size=0.3)

dtr.fit(train\_x, train\_y)

pred = dtr.predict(test\_x)

r2\_score(test\_y, pred)

next\_Year = df1[df1['Year']==2020][['Year', 'Month']]

next\_Year.Year.replace(2020,2021, inplace=True)

next\_Year= pd.get\_dummies(next\_Year)

temp\_2021 = dtr.predict(next\_Year)

temp\_2021 = {'Month':df1['Month'].unique(), 'Temprature':temp\_2021}

temp\_2021=pd.DataFrame(temp\_2021)

temp\_2021['Year'] = 2021

temp\_2021

next\_Year = df1[df1['Year']==2021][['Year', 'Month']]

next\_Year.Year.replace(2021,2022, inplace=True)

next\_Year= pd.get\_dummies(next\_Year)

temp\_2022 = dtr.predict(next\_Year)

temp\_2022 = {'Month':df1['Month'].unique(), 'Temprature':temp\_2022}

temp\_2022=pd.DataFrame(temp\_2022)

temp\_2022['Year'] = 2022

temp\_2022

forecasted\_temp = pd.concat([df1,temp\_2021], sort=False).groupby(by='Year')['Temprature'].mean().reset\_index()

fig = go.Figure(data=[

    go.Scatter(name='Yearly Mean Temprature', x=forecasted\_temp['Year'], y=forecasted\_temp['Temprature'], mode='lines'),

    go.Scatter(name='Yearly Mean Temprature', x=forecasted\_temp ['Year'], y=forecasted\_temp['Temprature'], mode='markers')

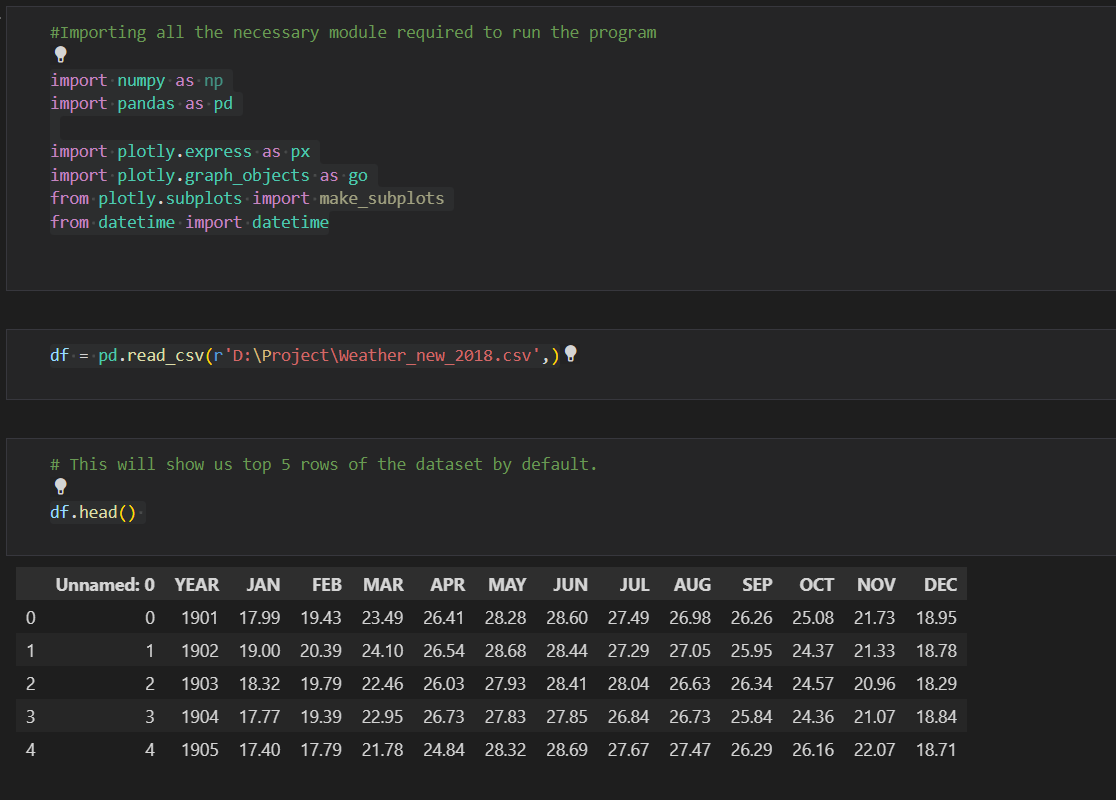
])

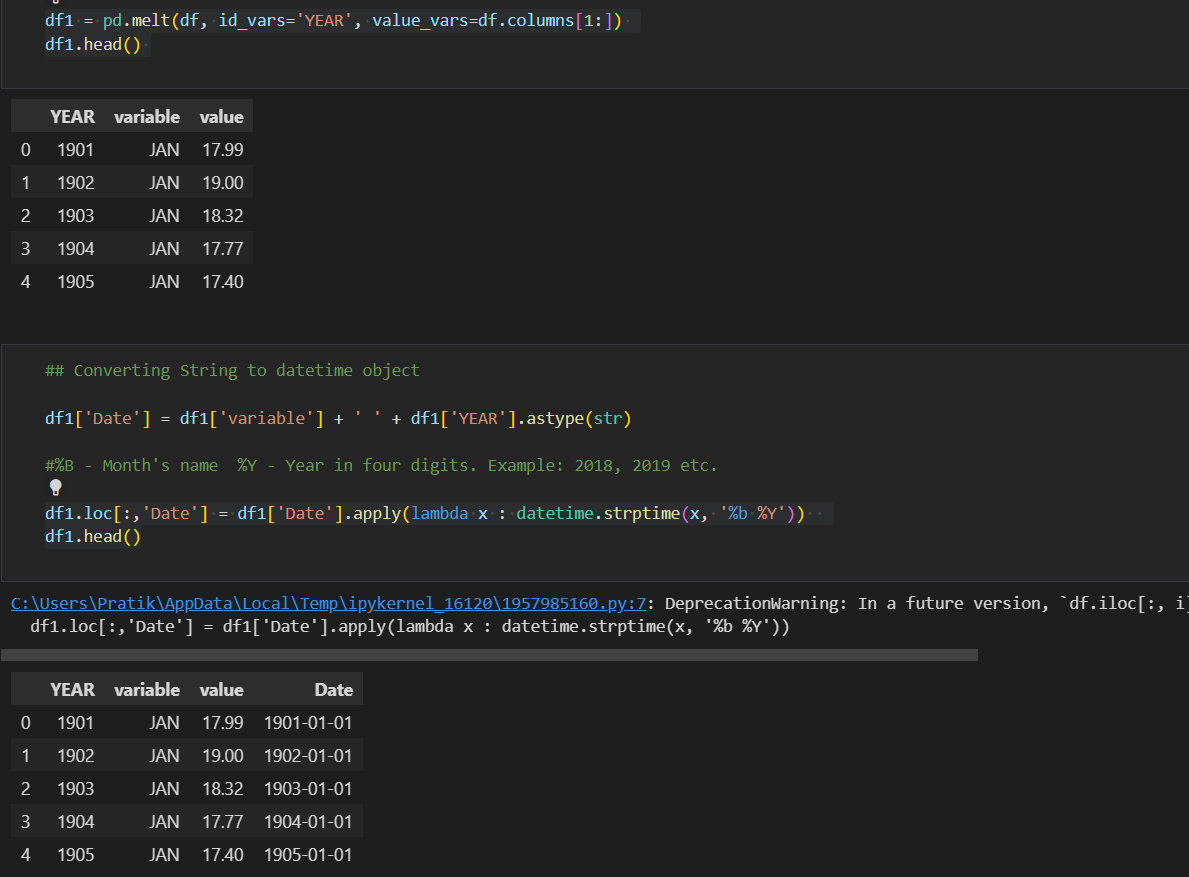
fig.update\_layout(title='Forecasted Temprature:',

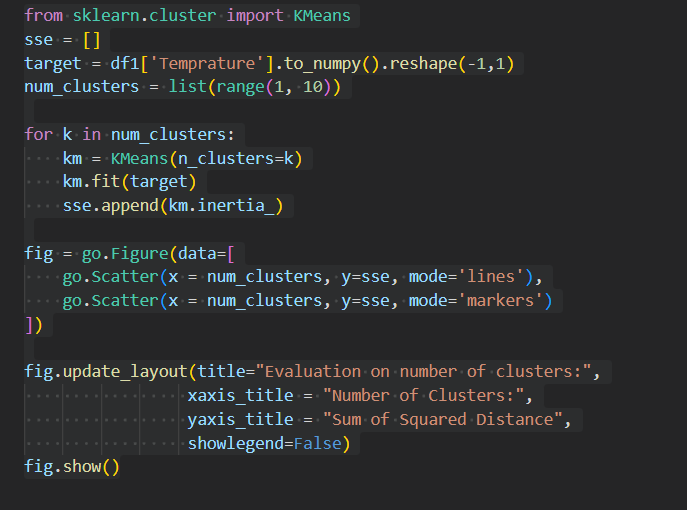
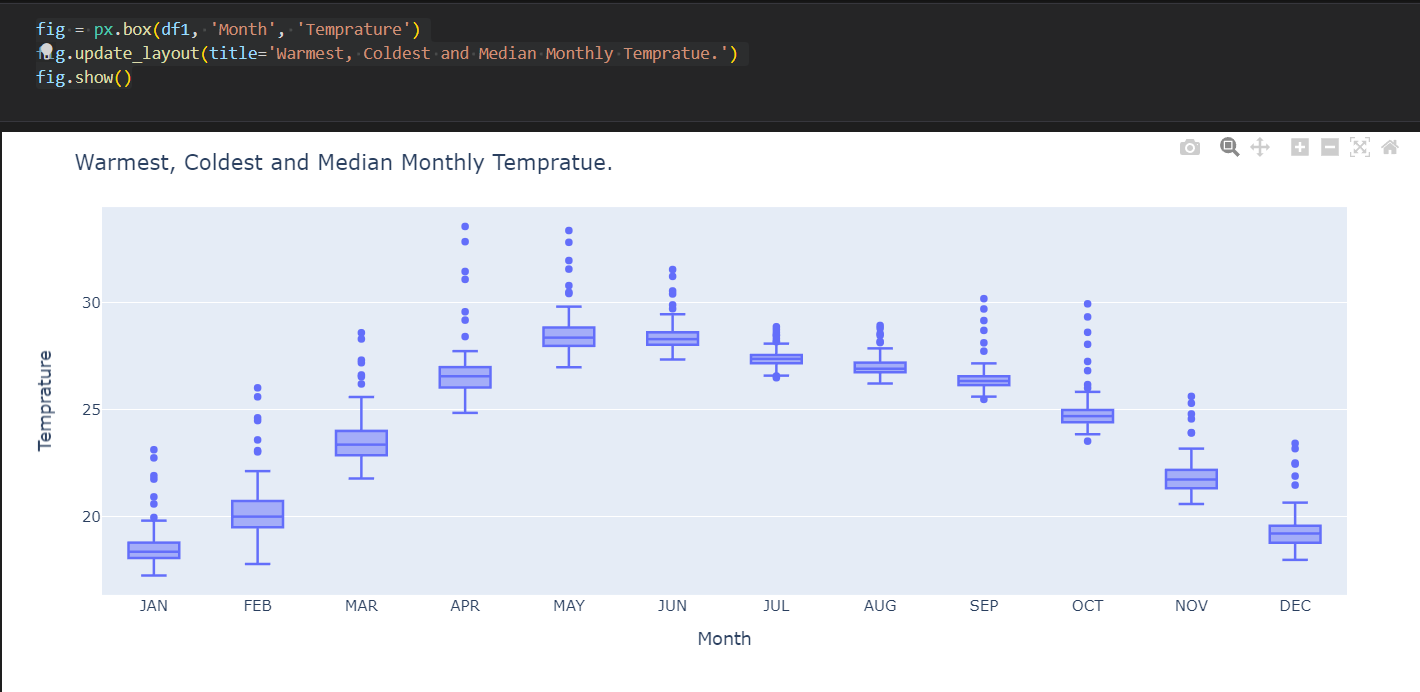
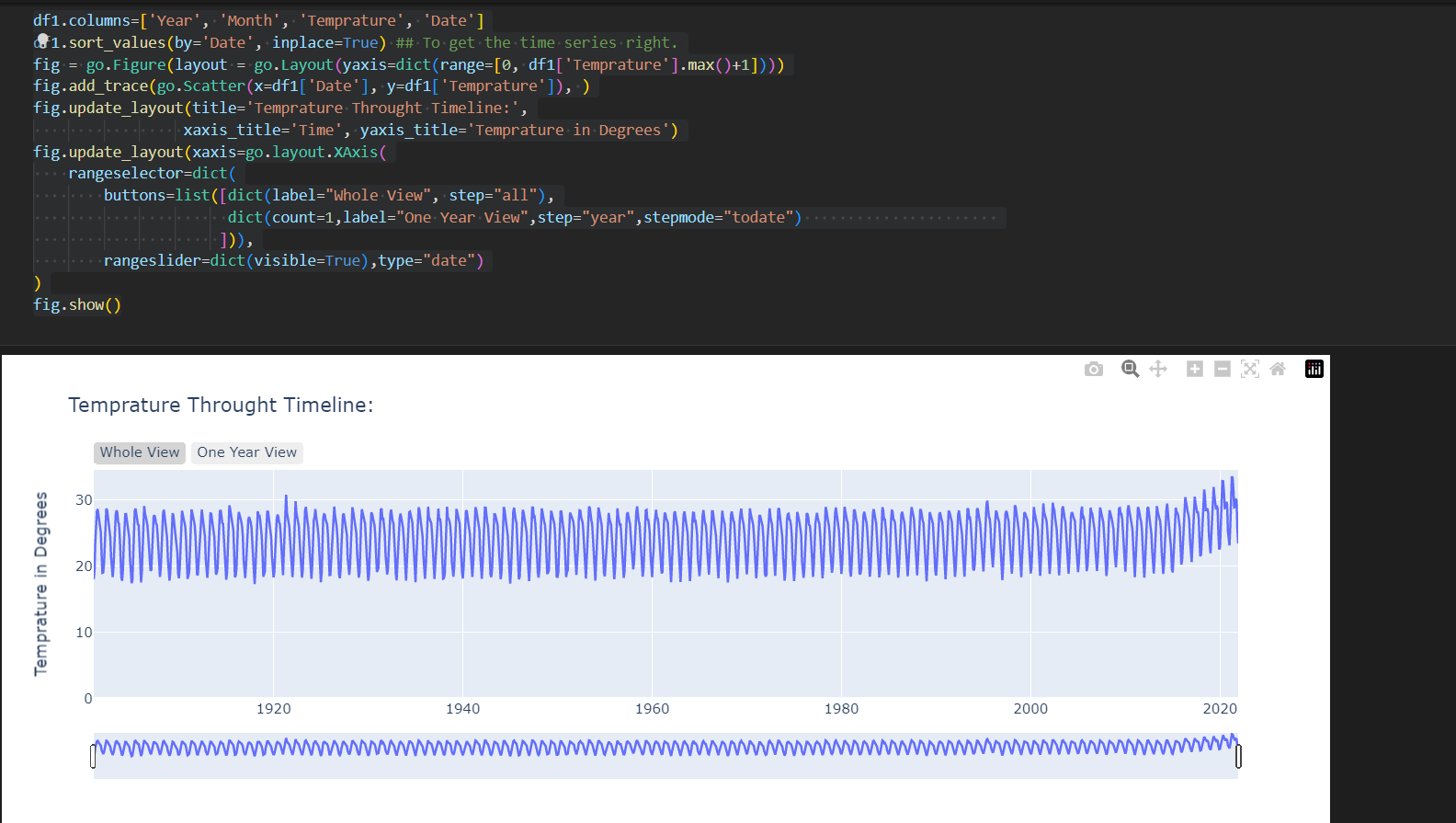
                 xaxis\_title='Time', yaxis\_title='Temprature in Degrees')

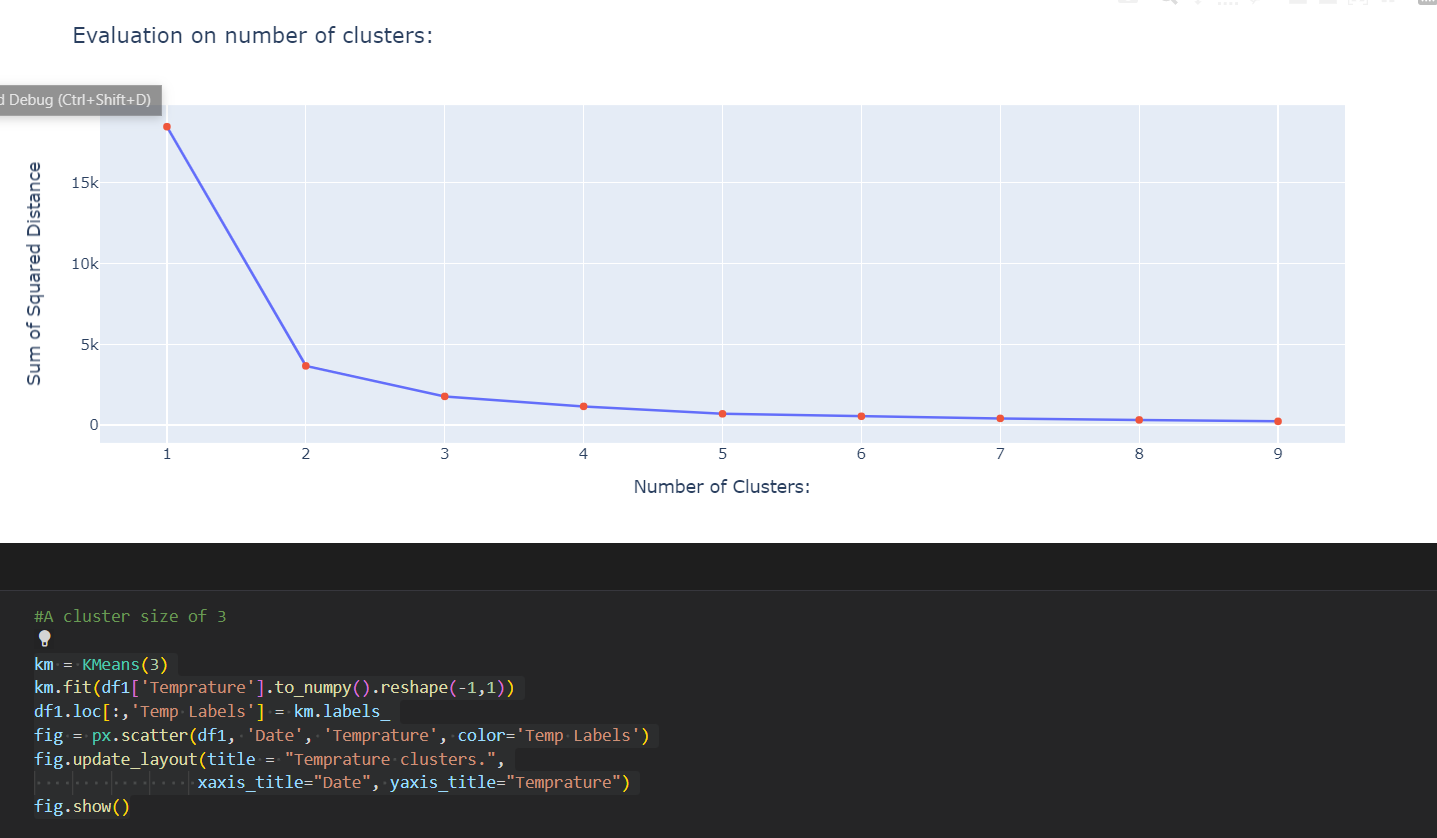
fig.show()

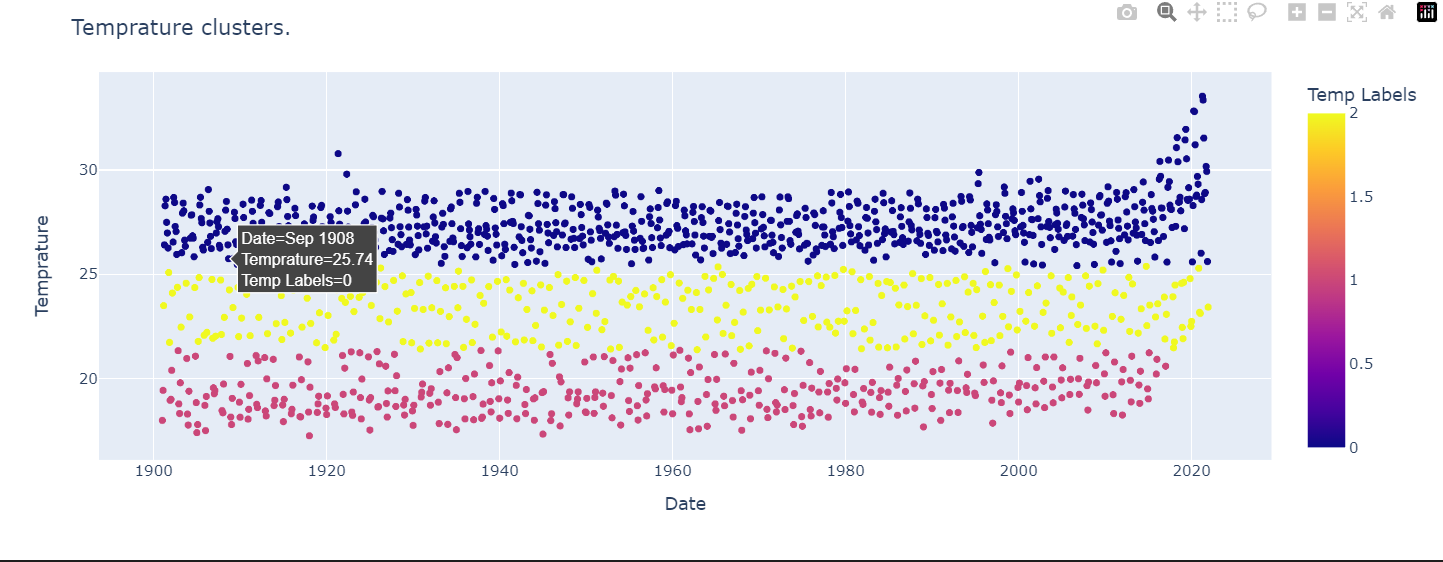
1. **Screen Layout and Result Layout**

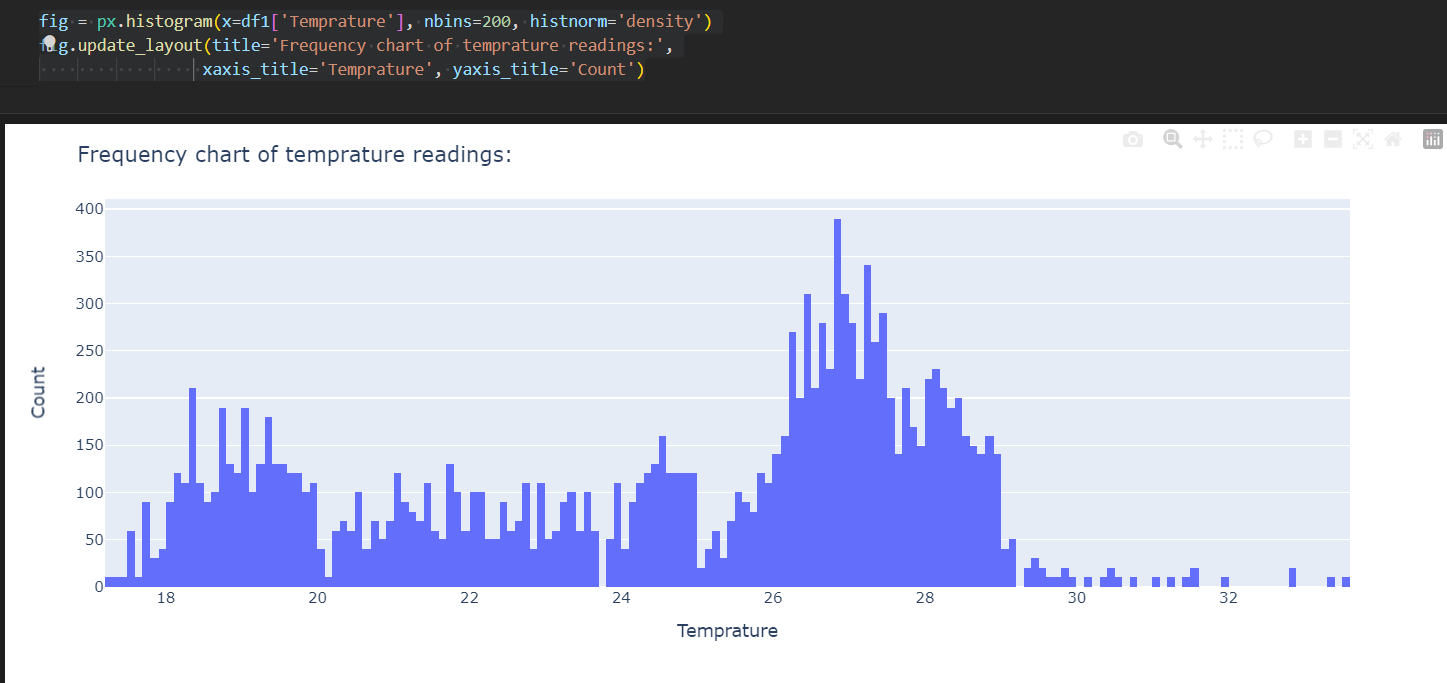
****

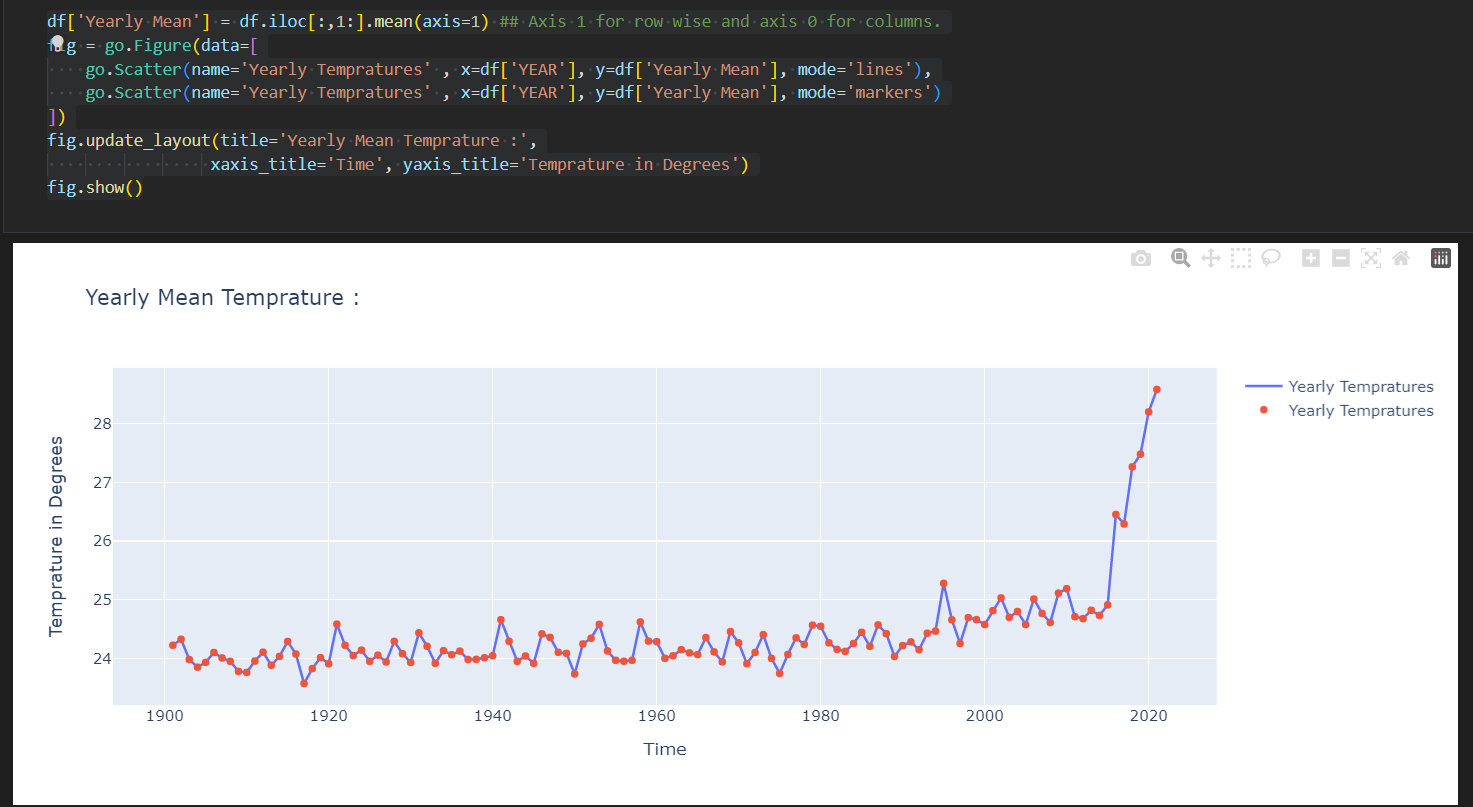
****

****

****

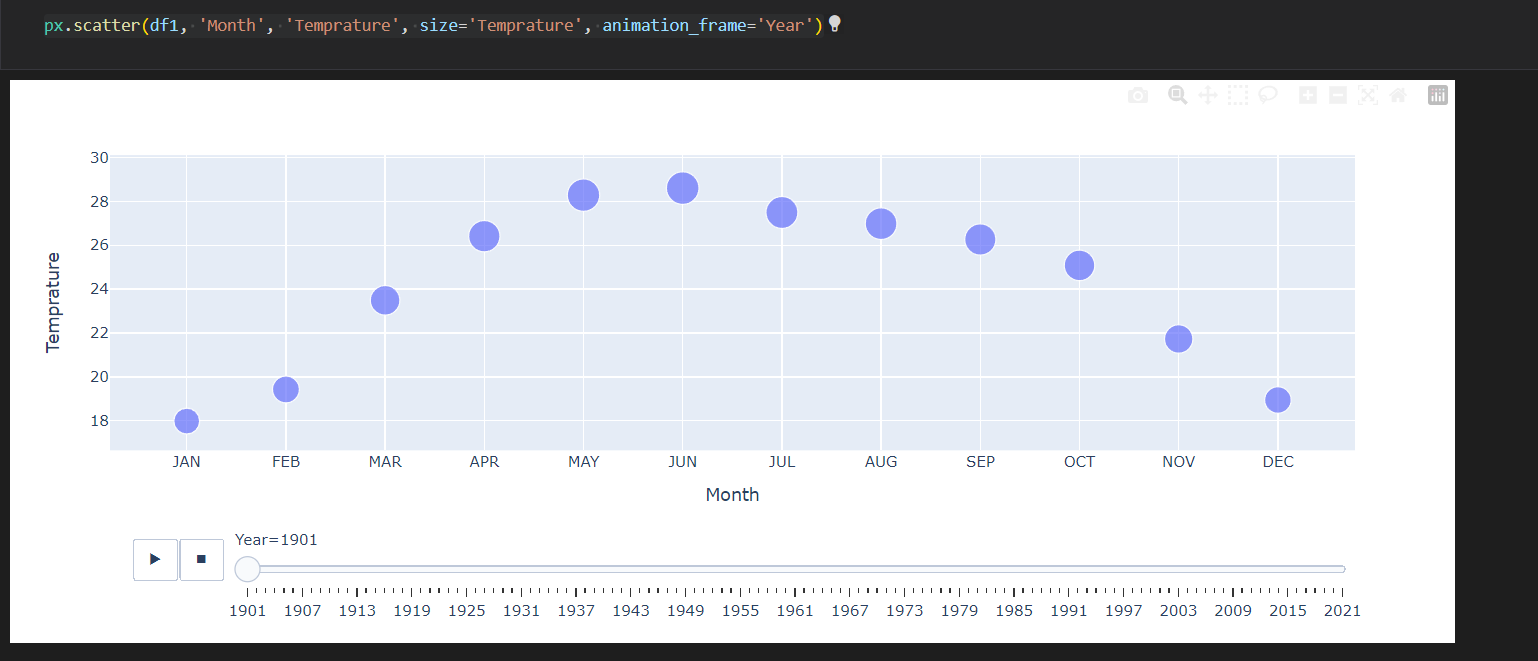
****

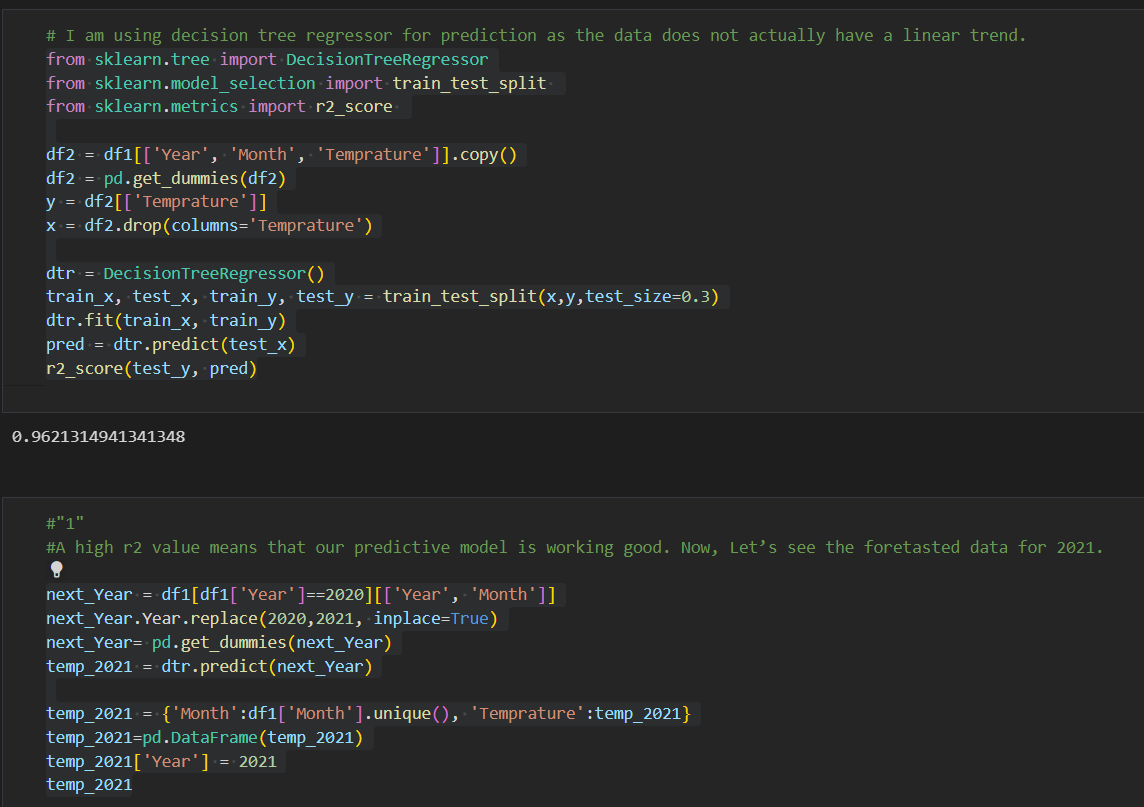
****

****

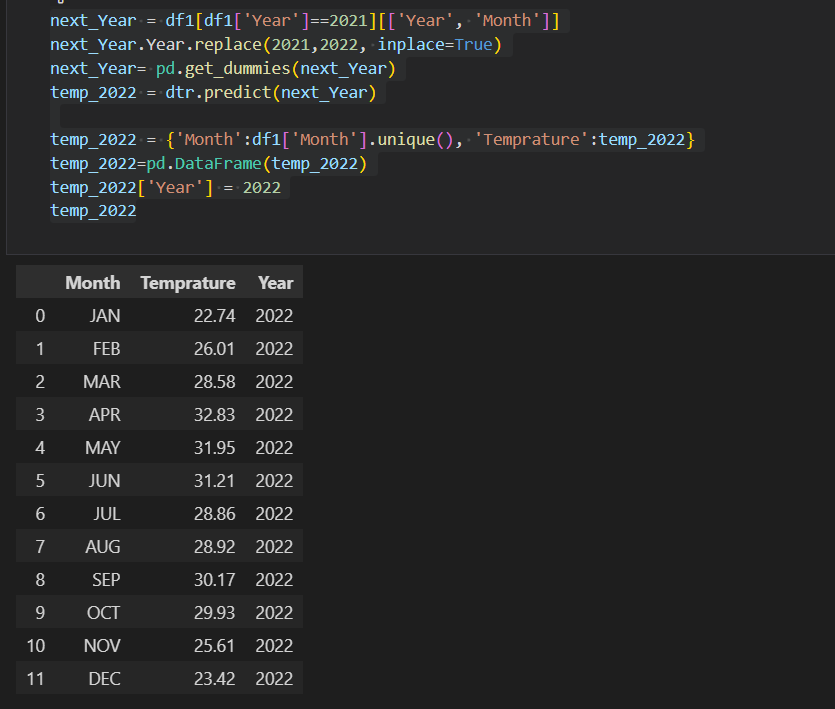
****

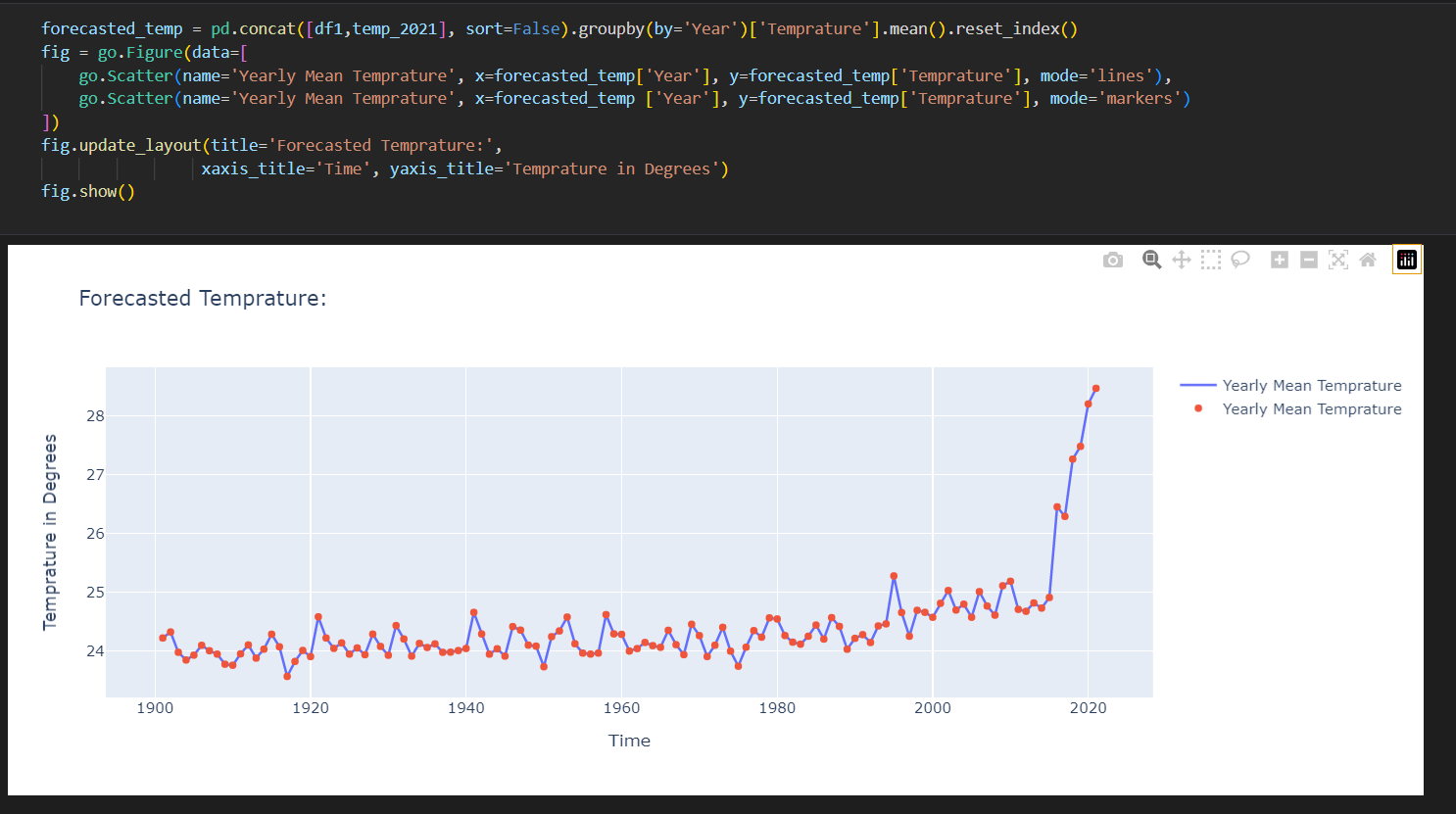
****

****

****

****

****

****

1. **Conclusion:**

This Program is built using Python and Machine Learning algorithm that receives and analyzes input data to predict output values based on historical data.

We have used the weather dataset that is available in the Kaggle website

We are successfully able predict the future weather using historical data

System maintenance:

* No Maintenance is required for this program the program train itself with the data and predicts the future based on the past data

1. **Future Scope:**

The System has adequate scope for modification in future if it is necessary.

The prediction accuracy can be increased by taking the model and training it with more data.

Creating a new optimized algorithm to optimize and increase the accuracy of the model.

1. **References**

**Full URL of online references:**

* + - * <https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/>
      * <https://en.wikipedia.org/wiki/Long_short-term_memory#:~:text=Long%20short-term%20memory%20(LSTM)%20is%20an%20artificial%20recurrent,the%20field%20of%20deep%20learning.&text=Relative%20insensitivity%20to%20gap%20length,learning%20methods%20in%20numerous%20applications>.
      * <https://towardsdatascience.com/sequence-models-and-recurrent-neural-networks-rnns-62cadeb4f1e1#:~:text=Sequence%20models%20are%20the%20machine,Applications%20of%20Sequence%20Models>
      * <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/#:~:text=The%20batch%20size%20is%20a%20hyperparameter%20that%20defines%20the%20number,more%20samples%20and%20making%20predictions>.
      * <https://www.w3schools.com/python/matplotlib_pyplot.asp>
      * <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>
      * <https://www.youtube.com/watch?v=5dMXyiWddYs&ab_channel=MATLAB>
      * <https://www.youtube.com/watch?v=vmEHCJofslg&ab_channel=KeithGalli>
      * <https://www.youtube.com/watch?v=hSPmj7mK6ng&ab_channel=CharmingData>
      * <https://www.youtube.com/watch?v=GwIo3gDZCVQ&ab_channel=edureka%21>
      * <https://nbviewer.jupyter.org/urls/bitbucket.org/hrojas/learn-pandas/raw/master/lessons/01%20-%20Lesson.ipynb>

<https://economictimes.indiatimes.com/marketstats/pid-51,pageno-1,year-2012,sortorder-asc,sortby-CurrentYearRank.cms>