**WEATHER FORCASTING USING MACHINE LEARNING**

**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. Here this system will predict weather based on parameters such as temperature, humidity and wind. System will take this parameter and will predict weather from previous data in database.

System will take current parameters entered by the user and will compare the parameters with the data in database and will predict the Weather. The output is shown in the maps and also a bar graph that is plotted to show the predicted results.

**Technology**

**Python:**

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular. Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-oriented way or a functional way.

**Machine learning:**

Machine Learning is making the computer learn from studying data and statistics. Machine Learning is a step into the direction of artificial intelligence (AI). Machine Learning is a program that analyses data and learns to predict the outcome. Machine Learning is used anywhere from automating mundane tasks to offering intelligent insights, industries in every sector try to benefit from it. You may already be using a device that utilizes it. For example, a wearable fitness tracker like Fitbit, or an intelligent home assistant like Google Home. But there are much more examples of ML in use.

* **Prediction:** Machine learning can also be used in the prediction systems. Considering the loan example, to compute the probability of a fault, the system will need to classify the available data in groups.
* **Image recognition:** Machine learning can be used for face detection in an image as well. There is a separate category for each person in a database of several people.
* **Speech Recognition:** It is the translation of spoken words into the text. It is used in voice searches and more. Voice user interfaces include voice dialling, call routing, and appliance control. It can also be used a simple data entry and the preparation of structured documents.
* **Medical diagnoses:** ML is trained to recognize cancerous tissues.
* **Financial industry and trading:** companies use ML in fraud investigations and credit checks.

**Libraries and Modules requirement:**

**Libraries**

1. **NumPy**

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Using NumPy, a developer can perform the following operations –

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

1. **Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

1. **Matplotlib**

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

1. **Sklearn**

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.

1. **Scatter**

Scatter plots are used to observe relationship between variables and uses dots to represent the relationship between them. The scatter () method in the matplotlib library is used to draw a scatter plot. Scatter plots are widely used to represent relation among variables and how change in one affects the other.

1. **KMeans**

The k-means clustering method is an unsupervised machine learning technique used to identify clusters of data objects in a dataset. There are many different types of clustering methods, but k-means is one of the oldest and most approachable. These traits make implementing k-means clustering in Python reasonably straightforward, even for novice programmers and data scientists.

1. **OpenWeatherMap API**

OpenWeatherMap is an online service, owned by OpenWeather Ltd, that provides global weather data via API, including current weather data, forecasts, nowcasts and historical weather data for any geographical location. The company provides a minute-by-minute hyperlocal precipitation forecast for any location.

**Modules**

1. **Weather Forecast:** In this module, system will take current parameters entered by the csv file and will compare the parameters with the data and will predict the weather.
2. **Visualization of predicted result:** This gives a picturized representation of the predicted weather in the form graphs.

**Methodology**

The process we have briefed in earlier section can be depicted pictorially and which is self- explanatory. We can divide our process in two modules namely:

1. **Weather Mining**

We have collected weather data from WORLD DATA CENTER for climate, Hamburg. We have decided to use NWS API for data collection in future.

Data formatting and cleaning: We have converted our data from .NC (netcdf) format to .CSV (comma-separated values) format because WEKA supports .CSV format.

1. **Recommendation**

For Project 1 we have to import data from CSV file with the help of python codes and after that we have to plot all the temperature graphs with the help of machine learning libraries and modules.

And For Project 2 we have to use Openweathermap API for that we have to register on Openweathermap site and get own API and after that we have to use it in the python code and get all the require fields we want.

**Hardware & software requirements**

1. **Hardware Requirements:**

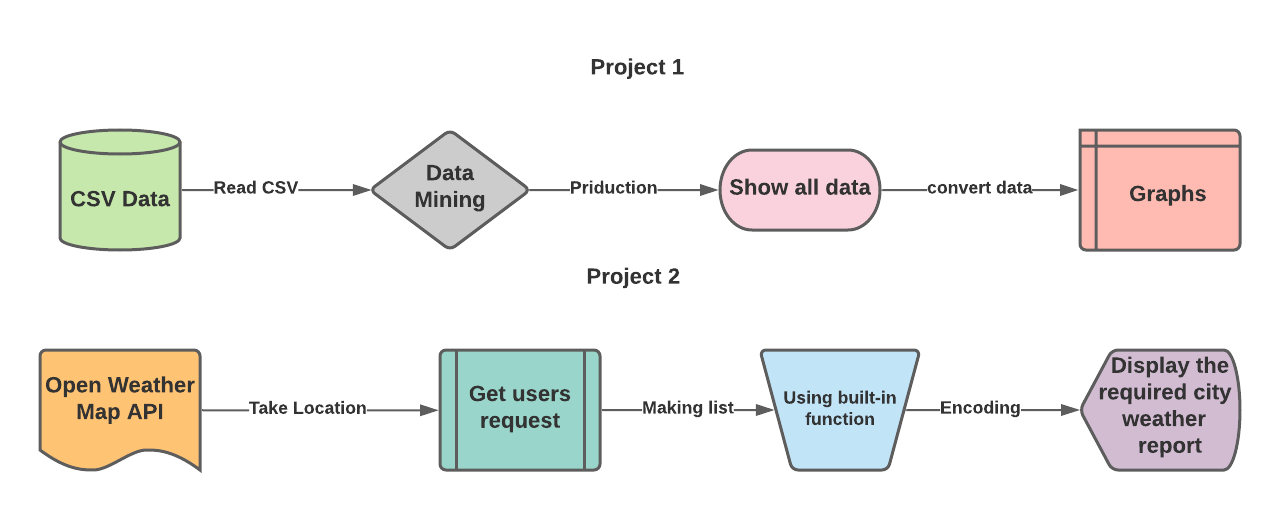
The hardware requirements listed below are almost in a significantly higher level which represents the ideal situations to run the system. Following are the system hardware requirements used:

1. Processor - Pentium –III
2. Speed - 1.1 GHz
3. RAM - 256 MB (min)
4. Hard Disk - 20 GB
5. Key Board - Standard Windows Keyboard
6. Mouse - Two or Three Button Mouse
7. **Software Requirements:**
8. Operating System – Windows
9. Python – 3.6 or above (also work with below versions)
10. Environment – Jyupter Notebook or Google Colab
11. Openweathermap API

**Data requirements**

We have collected weather data from WORLD DATA CENTER for climate, Hamburg. We have decided to use OpenWeatherMap API for data collection in future. We have converted our data from .NC (netcdf) format to .CSV (comma-separated values) format.

**Data Flow Diagram**

****

**Final Codes**

**# Project 1 Code**

# We are using decision tree regressor for prediction as the data does not actually have a linear trend.

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']==2022][['Year', 'Month']]

next\_Year.Year.replace(2022,2023, inplace=True)

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

temp\_2023 = dtr.predict(next\_Year)

temp\_2023 = {'Month':df1['Month'].unique(), 'Temprature':temp\_2023}

temp\_2023=pd.DataFrame(temp\_2023)

temp\_2023['Year'] = 2023

temp\_2023

forecasted\_temp = pd.concat([df1,temp\_2023], 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()

**# Project 2 Code**

import requests

from datetime import datetime

api\_key = '6f89b1f3c8543ded006c403c675457b9'

location = input("Enter the city name: ")

complete\_api\_link = "https://api.openweathermap.org/data/2.5/weather?q="+location+"&appid="+api\_key

api\_link = requests.get(complete\_api\_link)

api\_data = api\_link.json()

#create variables to store and display data

temp\_city = ((api\_data['main']['temp']) - 273.15)

weather\_desc = api\_data['weather'][0]['description']

hmdt = api\_data['main']['humidity']

wind\_spd = api\_data['wind']['speed']

date\_time = datetime.now().strftime("%d %b %Y | %I:%M:%S %p")

print ("Weather Stats for - {}  || {}".format(location.upper(), date\_time))

print ("Current temperature is: {:.2f} deg C".format(temp\_city))

print ("Current weather desc  :",weather\_desc)

print ("Current Humidity      :",hmdt, '%')

print ("Current wind speed    :",wind\_spd ,'kmph')

# making a list so that i can print the info to a text

txtlist = [temp\_city,weather\_desc,hmdt,wind\_spd,date\_time]

#using open() buit-in function to write to a text file

with open("textfile.txt" , mode= 'w' ,encoding= 'utf-8') as f :

#encoding = utf-8 for linux and cp1252 for win

    f.write("Weather Stats for - {}  || {}".format(location.upper(), date\_time))

    f.write("Current temperature is: {:.2f} deg C\n".format(txtlist[0]))

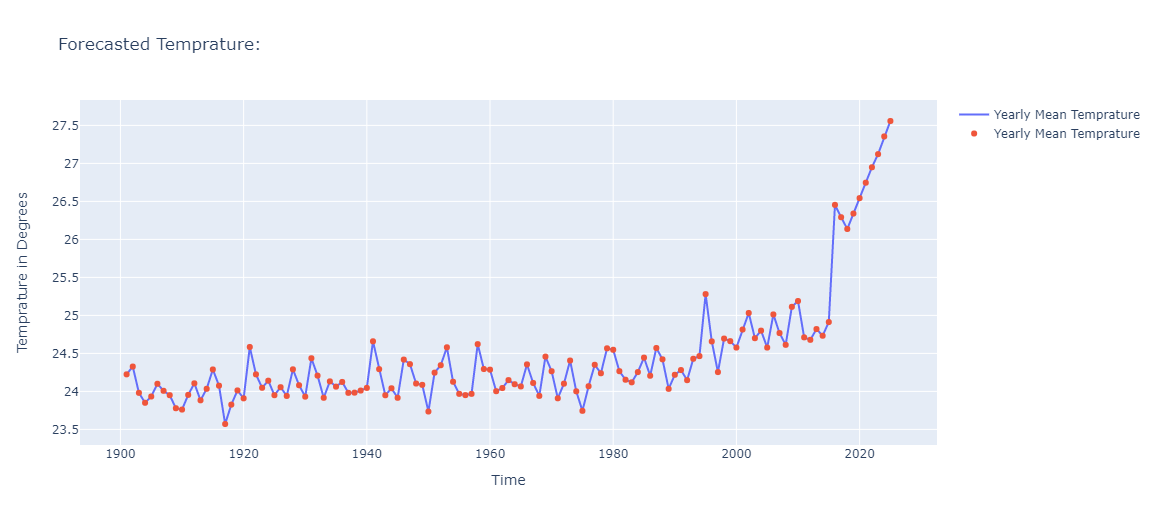
    f.write("{},{} \n".format("Current weather desc  :" ,txtlist[1]))

    f.write("{},{},{} \n".format("Current Humidity      :",txtlist[2],"%"))

    f.write("{},{},{} \n".format("Current wind speed    :",txtlist[3],"kmph"))

**Results**

**Project 1**



**Project 2**

Enter the city name: Chandigarh

Weather Stats for - CHANDIGARH || 20 Jan 2022 | 06:23:50 AM

Current temperature is: 14.19 deg C

Current weather desc : light rain

Current Humidity : 95 %

Current wind speed : 5.11 kmph

**Conclusion**

Traditionally, weather forecasting has always been performed by physically simulating the atmosphere as a fluid and then the current state of the atmosphere would be sampled. In the previous system the future state of the atmosphere is computed by solving numerical equations of thermodynamics. But this model is sometimes unstable under disturbances and uncertainties while measuring the initial conditions of the atmosphere. This leads to an incomplete understanding of the atmospheric processes, so it restricts weather prediction. Our proposed solution of using Machine learning for weather predicting is relatively robust to most atmospheric disturbances when compared to traditional methods. Another advantage of using machine learning is that it is not dependent on the physical laws of atmospheric processes. In the long run weather prediction using Machine Learning has a lot of advantages and thus it should be used globally.