 DOCUMENTATION

**WEATHER PREDICTION**

**(DATA SCIENCE)**

**CONTENT: -**

1. INTRODUCTION
2. OBJECTIVES
3. DATASET IMPORT
4. DATA EXPLORATION
5. DATA COLLECTION
6. PREPROCESSING
7. HYPOTHESIS FORMULATION
8. STATISTICAL ANALYSIS
9. CONCLUSION

**INTRODUCTION: -**

Weather forecasting is a complex task with significant implications for various industries, including agriculture, transportation and emergency response. Traditional methods are based on numerical weather models, but the development of information technology offers new opportunities to improve the accuracy of forecasts.

Big data technologies have made processing massive data sets produced by satellites, weather stations and other sources easier. Extensive data analysis enables real-time processing, revealing complex patterns and trends contribute to more accurate and timely weather forecasts.

**OBJECTIVES: -**

* Develop a data pipeline to collect and preprocess historical weather data.
* Explore and analyse the features affecting weather patterns.
* Implement machine learning models for weather prediction.
* Evaluate the performance of the models against traditional forecasting methods.

**Bringing Libraries in:**

A screenshot of a computer

Description automatically generated

**Pandas** for working with data.

For numerical operations, we are using **NumPy**.

The dataset is split into training and testing sets using the train\_test\_split function from **sklearn.model\_selection**.

For feature scaling, use **StandardScaler** from **sklearn. Preprocessing**.

Using **TensorFlow**, neural networks may be constructed and trained.

**TensorFlow, keras. Sequential, Dense, LSTM, and Dropout layers** are used to construct the neural network architecture.

Datetime and Time Delta for date-related operations.

For plotting, use **matplotlib.pyplot**.

**Bringing the Dataset Up:**

A Pandas DataFrame named data is filled with data from a CSV file called "modified\_file.csv" by the script.

**Treating Missing Data**

Firstly, we identified the missing values. Then, we identified the range of the column and used a random function to assign values to null rows within that range.

**TRAINING AND TESTING:-**

The data is being prepared in this area of your script for model testing and training:

**Choosing Features:**

A list of feature names is generated from the columns of df\_weather\_numeric by features = df\_weather\_numeric.columns. The parts you wish to utilise for testing and training are expected to be on this list.

**Data division:**

To divide the data into training and testing sets, use the train\_test\_split function.

The feature data for training and testing are located in x\_train\_weather and x\_test\_weather, respectively.

Assuming that **'TAVG'** is the target variable, the associated target values are in **y\_train\_continuous** and **y\_test\_continuous**.

**Data Standardisation:**

The feature data is standardised using **StandardScaler**.

Using the training data, scaler\_weather.fit(x\_train\_weather) determines the mean and standard deviation.

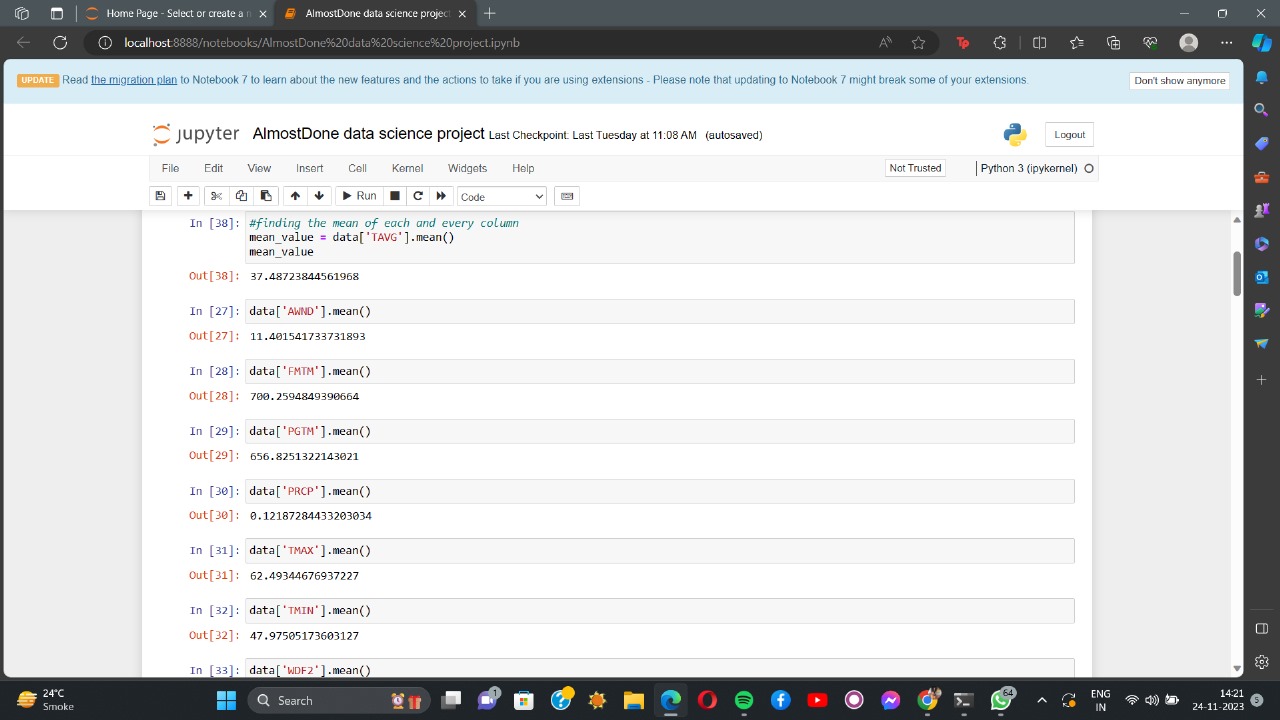
The training data is normalised using the formula x\_train\_weather = scaler\_weather.fit\_transform(x\_train\_weather).

The mean and standard deviation determined by x\_test\_weather = scaler\_weather.transform(x\_test\_weather) are used.

**DATA EXPLORATION: -**

Since weather conditions are variable and changeable, data analysis is critical in building adaptive forecasting models. Machine learning algorithms in the analytics toolkit can learn from historical data to identify emerging patterns and refine models’ ability to adapt to dynamic atmospheric conditions. Ultimately, the importance of data analysis in the context of meteorological datasets lies in its ability to unlock the potential of all available information, paving the way for more accurate and responsive weather forecasting in an ever-changing environment.

**MEAN: -**

****

It is the sum of the collection of numbers divided by the number of digits in the group.

The mean average **wind speed** is **11.401.**

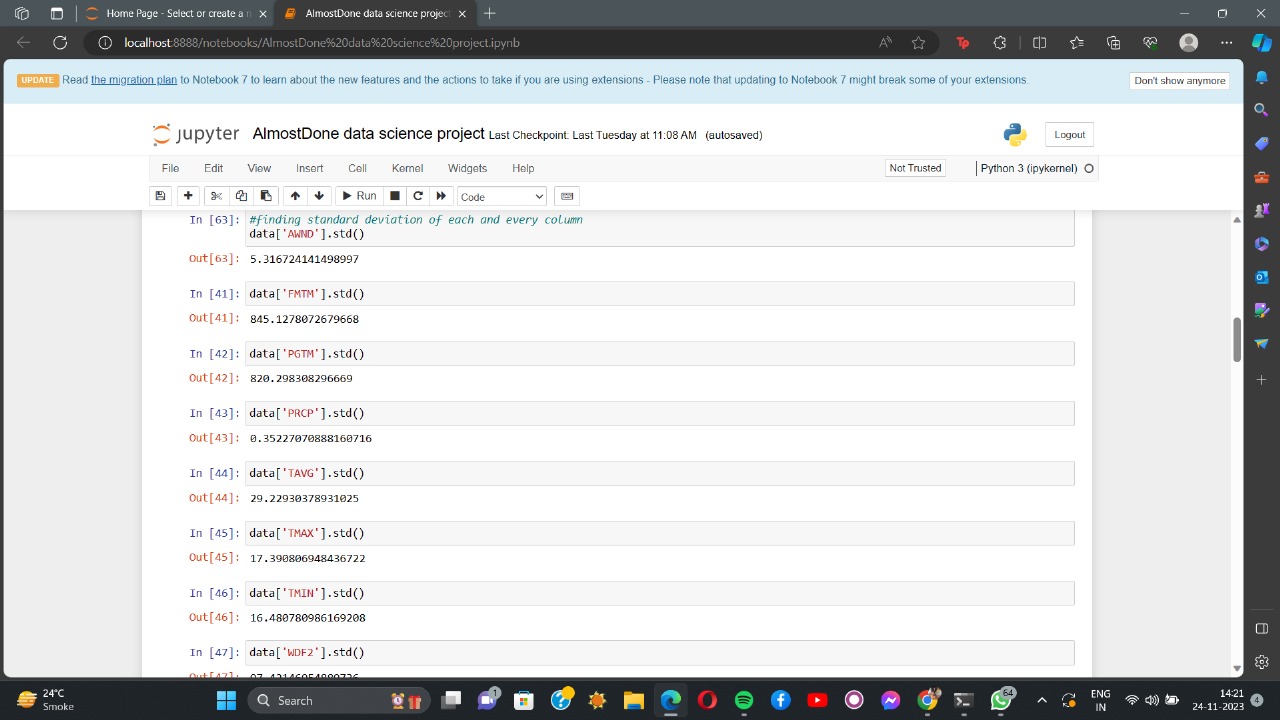
The mean **precipitation rate** is **0.1218.**

The mean average of the **maximum temperature** is **62.493.**

The mean average of the **minimum temperature** is **47.97.**

The mean **daily average temperature** is **37.48.**

**MEDIAN: -**

****

The median is the point that divides a data sample, population, or probability distribution into two halves—one higher and one lower. It can be understood as a dataset's central or "middle" value.

The median average **wind speed** is **10.51.**

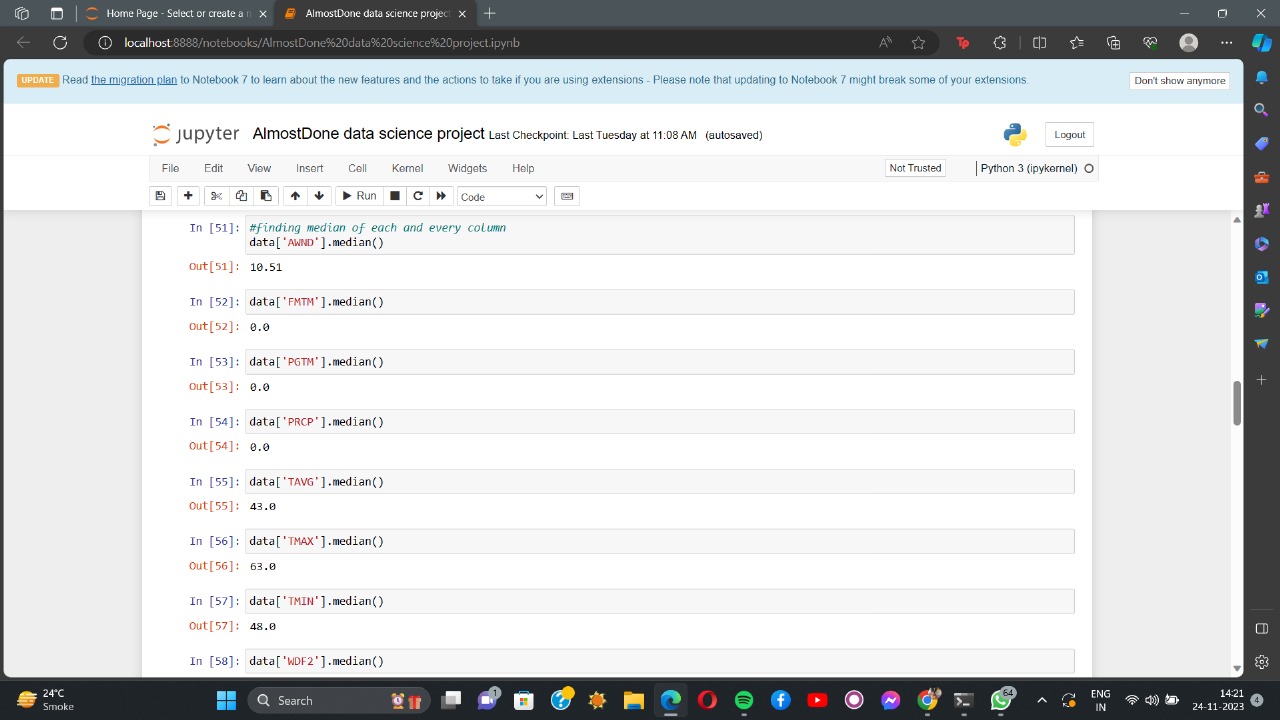
The median **precipitation rate** is **0.0.**

The median average of the **maximum temperature** is **63.**

The median average of the **minimum temperature** is **48.**

The median **daily average temperature** is **43.**

**STANDARD DEVIATION: -**

****

The **standard deviation** is the average amount of [variability](https://www.scribbr.com/statistics/variability/) in your dataset. It tells you, on average, how far each value lies from the mean.

The standard deviation of **wind speed** is **5.316.**

The standard deviation of the **precipitation rate** is **0.352.**

The standard deviation of the **maximum temperature** is **17.39.**

The standard deviation of the **minimum temperature** is **16.480.**

The standard deviation of the **daily average temperature** is **29.229.**

**DATA COLLECTION AND PREPROCESSING: -**

We have chosen the data from <https://www.noaa.gov/> (NOAA) U.S. Department of Commerce. In this, we are studying data from 10 years. It predicts weather changes and alerts us for upcoming disasters due to climatic changes caused by global warming in recent decades.

**PREPROCESSING: -**

A screenshot of a computer

Description automatically generated

**'DATE'** Column to Numerical Format Conversion:

At first, the 'DATE' column is formatted as a date. The 'DATE' column is converted to a datetime format using pd.to\_datetime, with the input format specified as '%d-%m-%Y' (day-month-year).

Next, data['DATE']. The 'DATE' column is overwritten with these numerical day-of-year values after each date's day of the year is extracted using the function dt—day of the year.

**Remove All Non-Numeric Columns:**

The list of column names ('STATION' and 'NAME') that are non-numeric and might not be required for modelling is contained in non\_numeric\_columns.

Use data to make a new DataFrame (df\_weather\_numeric) without these non-numeric columns. drop(columns=non\_numeric\_columns).

**CREATING DATA SEQUENCES:-**

To turn your time series data into sequences for time series forecasting, you define a function called create\_sequences. This is a typical step when using recurrent neural networks (RNNs), such as LSTM, for time series prediction.

A screenshot of a computer

Description automatically generated

**Contributions:**

**Data**: The time series input data.

The target variable (e.g., 'TAVG') is stored in **target\_column**.

**Sequence\_length**: Every sequence's length.

**Results:**

**Sequences**: A collection of input sequences with sequence\_length time steps for each line.

**Targets**: For every sequence, an array containing the matching target values.

**Handling:**

The function generates sequences of length sequence\_length by iterating through the provided data.

It extracts the associated target value (label) for each sequence, which is the value at the time step after the series.

It adds each sequence to the lists of targets and lines accordingly.

**Reply:**

The function returns the targets and sequences as NumPy arrays.

When you wish to organise your time series data to train a model that forecasts future values based

**LSTM MODEL: -**

A screenshot of a computer

Description automatically generated

**Model Sequential:**

A sequential model is initialised using model\_time\_series = Sequential().

**Layer LSTM:**

time\_series\_model. Relu activation function, 50 units (neurons) in the LSTM layer, and an input shape that corresponds to the shape of your input sequences (X\_train\_seq) are added by adding

(LSTM(50, activation='relu',input\_shape=(X\_train\_seq. Shape[1], X\_train\_seq. Shape[2]))).

**Layer of Dropout:**

A dropout layer with a dropout rate 0.5 is added using model\_time\_series.add(Dropout(0.5)). A regularisation method called dropout aids in preventing overfitting.

**Layer Dense (Output):**

Add model\_time\_series(Activation='linear', Dense(1)) incorporates a dense layer with a linear activation function and a single output neuron. This works well for regression tasks involving the prediction of a continuous variable.

**Gathering:**

The model is compiled using model\_time\_series.compile(optimizer='adam', loss='mean\_squared\_error', metrics=['mean\_absolute\_error']). It uses the Adam optimiser, mean absolute error as a monitoring parameter during training, and mean squared error as the loss function—a standard approach for regression issues.

A screenshot of a computer

Description automatically generated

**TRAINED MODEL OF DATASET:-**

Training the Model: Using your training sequences (X\_train\_seq) and matching targets (y\_train\_seq), model\_time\_series.fit(X\_train\_seq, y\_train\_seq, epochs=100, batch\_size=32, validation\_split=0.2) trains the model.

With a batch size 32, it runs for 100 epochs or iterations over the whole training dataset. Twenty per cent of the training data will be used for validation throughout training, according to the assurance split 0.2.

Saving the Trained Model (Optional): model\_time\_series.save('temperature\_time\_series\_model.h5') uses the Hierarchical Data Format version 5 (HDF5) format to store the trained model to a file with the name 'temperature\_time\_series\_model.h5'.

This is an optional step. However, using the trained model again without retraining can be helpful.

Remember to change the file name and path to suit your tastes.

**HYPOTHESIS FORMULATION: -**

Hypothesis Formulation:

1. **Effect of Weather Features on Temperature (TAVG): -**

- **Null Hypothesis (H0):** There is no significant relationship between the selected weather features (AWND, FMTM, PGTM, PRCP, TMAX, TMIN, WDF2, WDF5, WSF2, WSF5) and the average temperature (TAVG).

- **Alternative Hypothesis (H1):** A significant relationship exists between the selected weather features and the average temperature.

2. **Effect of Time on Temperature (TAVG): -**

- **Null Hypothesis (H0):** Time (day of the year) does not significantly impact the average temperature.

- **Alternative Hypothesis (H1**): Time (day of the year) significantly impacts the average temperature.

3. **Impact of Future Dates on Predictions: -**

- **Null Hypothesis (H0):** There is no significant change in the predicted average temperature when forecasting for future dates.

- **Alternative Hypothesis (H1):** Forecasted average temperatures for future dates significantly differ from the historical data, indicating a change in temperature patterns.

Using the code, these hypotheses form the basis for evaluating the significance of relationships and capabilities in weather forecasting. Statistical tests and analyses can be conducted to either accept or reject these null hypotheses based on the results obtained during the exploration and experimentation with the data and model.

**STATISTICAL ANALYSIS: -**

Numerical weather prediction (NWP) plays a significant role in modern weather forecasting. NWP uses mathematical models that simulate the behaviour of the atmosphere based on the fundamental laws of physics. These models divide the atmosphere into three-dimensional grids and solve equations to predict future weather conditions. NWP models consider variables such as temperature, humidity, wind speed and direction, and pressure to generate forecasts.

Machine learning algorithms have also become an integral part of weather prediction. Machine learning models can predict future weather conditions by analysing historical weather data and identifying patterns. These algorithms use regression, classification, and clustering techniques to process large amounts of data and extract meaningful insights. They can improve the accuracy and reliability of short-term weather forecasting.

Mean value of **'TAVG': 37.482**

Mean value of **'AWND': 11.401**

Mean value of **'FMTM': 700.259**

Mean value of **'PGTM': 656.825**

Mean value of **'PRCP': 0.121**

Mean value of **'TMAX': 62.493**

Mean value of **'TMIN': 47.975**

Mean value of **'WDF2': 215.671**

The mean value of **'WDF5': 213.671**

Mean value of **'WSF2': 22.057**

Mean value of**'WSF5': 27.095**

**CONCLUSION: -**

Based on my analysis of the available information, I have concluded that weather prediction is a complex yet crucial task that significantly impacts various aspects of our lives and the economy.

* To create climate-quality data from satellite information, it is essential to ensure reliable and long-term observations involving user communities and other agencies to enhance and expand community involvement in generating climate data records.
* Data science can be crucial in improving short-term weather prediction using machine learning algorithms.
* Therefore, my project on weather prediction using data science will focus on utilising machine learning algorithms to improve short-term weather forecasting accuracy and reliability.

**The Actual Prediction**

