Time Series Analysis of Weather Data

Ginu Varghese
Department of Computing
Science and Mathematics
Dundalk Institute of Technology
Dundalk, Co.Louth, Ireland
d00251842@student.dkit.ie

***Abstract*— The Weather forecasting is considered as one of the most challenging problems around the world. There are various reasons for this and one of the reasons is its tested values in meteorology. The structure of climate is controlled by the changes in rainfall, humidity and temperature and the weather can be analyzed by time series analysis of these factors. This paper describes the time series analysis of the daily hour temperature data from 2009 to 2011 using the ARIMA model.**

Keywords—Time series analysis, ARIMA, temperature, weather.

I. INTRODUCTION

Weather refers to the day-to-day atmospheric conditions including the temperature, precipitation, humidity. Most of our daily decisions are impacted by the weather conditions. The first thing most people do when they wake up is to check the weather conditions to plan the rest of the day. In general, the daily weather conditions can affect the feelings and insight of people towards their life and world. So, it is important to analyze and predict the patterns of weather for the easy going of life[1].

The main components of weather are temperature, wind, humidity, and atmospheric pressure. Temperature refers to how cold or hot the atmosphere is, and the humidity is the quantity of water vapor in the atmosphere usually referred as the atmospheric moisture. Atmospheric pressure is the pressure within the Earth's atmosphere[2]. These components can be used to describe weather at any time. This data with change in time can be used by meteorologists to find out the weather conditions and to forecast the weather patterns for the future. It also helps the people to use the daily forecast to plan their day-to-day activities and life according to the weather patterns of the day.

The weather dataset for the project was downloaded from the Harvard Dataverse. The selected dataset was available to the public for download and use with proper citation to the site. The dataset gives the information about the temperature, humidity, and pressure along with the wind direction of the day and the weather condition of the recorded hours in each day from 2009 to 2011.

Weather forecasting is a challenging are of investigation for the scientists and it is one of the most important types of forecasting because the agriculture sector and many industries as well as the population are highly dependent on the weather conditions [3]. The weather forecasting is used to predict and warn about the various natural disasters that are caused by the changes in the atmosphere. The current technology is suing the statistical model-based weather forecasting methods which requires many computations. The objective of this project is to study the daily weather patterns of different variables such as temperature, pressure, and humidity from 2009 to 2011 and determine the behavior of data from time to time and to train models on the data and to learn how to forecast the weather data for future.

The technology used in the project is listed in the table below.

Technology
Programming language: Python (Jupyter Notebook, Google Colab)
Python libraries: • Pandas • Numpy • Matplotlib • Seaborn • scipy.stats • statsmodels

TABLE I. Technology used for the analysis

This paper is organized as follows: Section II describes the existing works. Time series analysis flow diagram is described in the Section III. The data description is mentioned in the Section IV. The summary statistics and visualizations are described in Section V. Section VI gives the research questions and methods used in the project. The results are described in the Section VII. Then Conclusion is given in Section VIII. Ethical considerations are mentioned in Section IX. And finally, References are mentioned in Section X.

II. Related Works

Geophrey Odero published a paper on the time series analysis of weather data in South Carolina based on the data from January 2003 to December 2017[1]. In this paper they examined the seasonality, trends and stationarity of the data and determined an appropriate model. They decomposed the average daily temperature time series data into a trend component, a seasonal component, and a random noise component to accurately understand the time series data. ARIMA model was used for the time series analysis and accuracy measures were used to evaluate the performance of the model.

Artificial Neural Network (ANN) and MATLAB was used to analyze and forecast the weather conditions by Neeraj Kumar in [4] to predict the maximum and minimum temperature. They used the mean squared error to evaluate the model and confirmed that the model based on multilayer perceptron can forecast weather successfully. They also concluded that this type of network can correctly provide the mapping between actual and the predicted value using the historical data.

A.K Shukla published a paper on weather forecasting based on the data from 1881 to 2007 for Uttarakhand, India. Mann-Kendall test statistics was used to analyze the trend of the weather data and the SARIMA model and winter's exponential smoothing model was used for forecasting the weather parameters. The study of forecast models showed that SARIMA model is the most efficient model for forecasting the monthly humidity, monthly maximum temperature, and monthly minimum temperature. The Winter's model was found to be the most efficient model for forecasting Monthly Humidity [5].

In a study by Pitshu Mulomba Mukadi[6], the series of two weather variables, the monthly temperature and monthly precipitation in Spain were analyzed from the 1940s to 2013. They used ARIMA models to represent their variation. In the paper they found significant trends in some of the temperature series, and precipitation trends in some of them. The results obtained from the trends were analyzed with other studies in different regions, to confirm whether the same trend was retained over time. The 12-month predictions were made by calculating the errors with the observed data using the ARIMA models. Different scenarios such as wildfires, agricultural crops, pests, and many diseases could all benefit from the predictions made with these models[6].

III. Time Series Analysis Flow Diagram

The different phases of this project are: the data collection, data preparation, data modelling and model evaluation. This is shown in Figure 1.

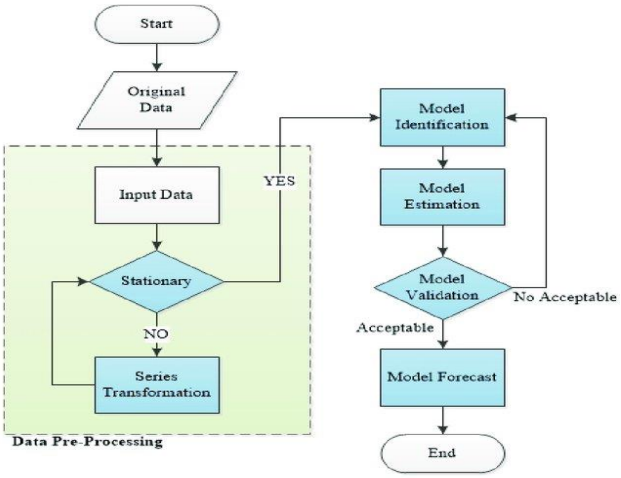


Figure 1. Flow diagram for Time series Analysis[7]

Source: https://www.researchgate.net/figure/Flowchart-of-the-time-series-analysis methodology_fig2_337404958

IV. DATA DESCRIPTION

The dataset describes the daily temperature, pressure, and humidity of each hour from 2009 to 2011. The temperature, humidity and pressure are recorded at each hour from 8 am to 11 pm. The dataset also contains the details about the condition, the wind speed, direction of wind and the air quality health index. The data was collected from the Harvard Dataverse. There are 9 variables and 9861 observations in the dataset. The time series analysis was performed for the temperature variable since it has no missing variables.

The data is collected from the Harvard Dataverse website, and it is available for downloading and use[8].

A. Types of variables

The TABLE II explains the type of variables in the dataset. There are 9 variables and 9861 observations in the dataset.

Types of variables			
Variable	**Category**	**Type**	**Description**
read_dt	Date	Datetime	Date in which the data is recorded.
read_tm	Continuous Numerical	Datetime	Time in which the data is recorded.
condition	Nominal Categorical	String	Weather condition of the day.
temperature	Continuous Numerical	Float	Temperature of the hour in degree Celsius.
pressure	Continuous Numerical	Float	Atmospheric Pressure of the hour in hPa.
humidity	Continuous Numerical	Float	Humidity of the hour.
wind_direction	Nominal Categorical	String	Wind direction of the hour.
wind_speed	Continuous Numerical	Float	Wind speed of that hour.
air_quality_health_index	Continuous Numerical	Float	Air quality health index.

TABLE II. Type of variables in the dataset

B. Cleaning of Data

The data was cleaned using the python library functions. The data type of variables was checked using the dtype() and info() functions. The duplicates and null values were checked. There were no duplicates in the dataset. There are some missing values in the columns other than temperature. These values can be dealt with backward filling or forward filling. But I have kept those values like that since I will be using the temperature column for analysis, and it has no missing values. The dataset is found to be balanced since it has 9861 rows and 9 features.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9861 entries, 0 to 9860
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   read_dt                  9861 non-null   object
 1   read_tm                  9861 non-null   object
 2   condition                9861 non-null   object
 3   temperature              9861 non-null   float64
 4   pressure                 9506 non-null   float64
 5   humidity                 9858 non-null   float64
 6   wind_direction           8657 non-null   object
 7   wind_speed               9861 non-null   float64
 8   air_quality_health_index  7722 non-null   float64
dtypes: float64(5), object(4)
memory usage: 693.5+ KB
```

Figure 2. Data Types of the variables using the info() function

V. Descriptive Statistics And Data Visualizations

This section includes the Exploratory data analysis of the dataset. The measures of central tendency and visualizations are also included in this section.

I. Summary Statistics

A. Date

There are 9861 total values in the date column with 371 unique values. The mode of the date variable is 2010-11-05 with a frequency of 40 observations.

```
dataset['read_dt'].describe()

count            9861
unique            371
top        2010-11-05
freq                40
Name: read_dt, dtype: object
```

Figure 3. Summary statistics of *read_dt* variable

B. Time

There are 9861 total values in the time column with 721 unique values. The mode of the time variable is 21:00:00 with a frequency of 369 observations.

```
dataset['read_tm'].describe()

count          9861
unique          721
top        21:00:00
freq            369
Name: read_tm, dtype: object
```

Figure 4. Summary statistics of *read_tm* variable

C. Temperature

There are 9861 total values in the temperature column with a mean of 10.76. The median of the temperature variable is 10.20. The minimum temperature value is 0 and the maximum is 29.20 degree Celsius. The IQR lies between 6.90 and 14.50 degree Celsius.

```
dataset['temperature'].describe()

count    9861.00
mean       10.76
std         5.20
min         0.00
25%         6.90
50%        10.20
75%        14.50
max        29.20
Name: temperature, dtype: float64
```

Figure 5. Summary statistics of *temperature* variable

D. Pressure

There are 9861 total values in the pressure column with a mean of 101.44. The median of the pressure variable is 101.50. The minimum pressure value is 98.30 hPa and the maximum is 103.90 hPa. The IQR lies between 101 and 102 hPa.

```
dataset['pressure'].describe()

count    9506.00
mean      101.44
std         0.82
min        98.30
25%       101.00
50%       101.50
75%       102.00
max       103.90
Name: pressure, dtype: float64
```

Figure 6. Summary statistics of *pressure* variable

E. Humidity

There are 9861 total values in the humidity column with a mean of 77.8. The median of the variable is 79. The minimum humidity is 26 and the maximum is 100. The IQR lies between 69 and 89.

```
dataset['humidity'].describe()

count    9858.0
mean       77.8
std        13.3
min        26.0
25%        69.0
50%        79.0
75%        89.0
max       100.0
Name: humidity, dtype: float64
```

Figure 7. Summary statistics of *humidity* variable

F. Wind Speed

There are 9861 total values in the wind speed column with a mean of 13.45. The median is 13. The minimum wind speed is 0 and the maximum is 78. The IQR lies between 8 and 18.

```
dataset['wind_speed'].describe()

count    9861.00
mean       13.45
std         9.23
min         0.00
25%         8.00
50%        13.00
75%        18.00
max        78.00
Name: wind_speed, dtype: float64
```

Figure 8. Summary statistics of *wind_speed* variable

G. Wind Direction

There are 9861 total values in the wind direction column with 17 unique values. The mode of the variable is value 'E' with a frequency of 2685.

```
dataset['wind_direction'].describe()

count     8657
unique      17
top          E
freq      2685
Name: wind_direction, dtype: object
```

Figure 9. Summary statistics of *wind_direction* variable

H. Condition

There are 9861 total values in the wind direction column with 21 unique values. The mode of the variable is value 'cloudy' with a frequency of 3026.

```
dataset['condition'].describe()

count       9861
unique        21
top       Cloudy
freq        3026
Name: condition, dtype: object
```

Figure 10. Summary statistics of *condition* variable

I. Air quality health index

There are 9861 total values in the air quality health index column with a mean of 2.27. The median is 2. The minimum value is 1 and the maximum is 6. The IQR lies between 2 and 3.

```
dataset['air_quality_health_index'].describe()

count    7722.00
mean        2.27
std         0.58
min         1.00
25%         2.00
50%         2.00
75%         3.00
max         6.00
Name: air_quality_health_index, dtype: float64
```

Figure 11. Summary statistics of *air_quality_health_index* variable

J. Skewness of Data

The skewness of the data is identified using the skew() function. The variables pressure and humidity are found to be left skewed.

```
dataset.skew()

temperature                0.36
pressure                  -0.68
humidity                  -0.54
wind_speed                 0.93
air_quality_health_index   0.82
dtype: float64
```

Figure 12. Skewness of data using skew() function

II. Visualizations

A. Histogram

Histogram is used to detect the skewness and the presence of outliers in the data and to identify whether the data distribution is Gaussian or not[9]. None of the variables are following complete gaussian distribution. The pressure and temperature variables are following a rough gaussian distribution.

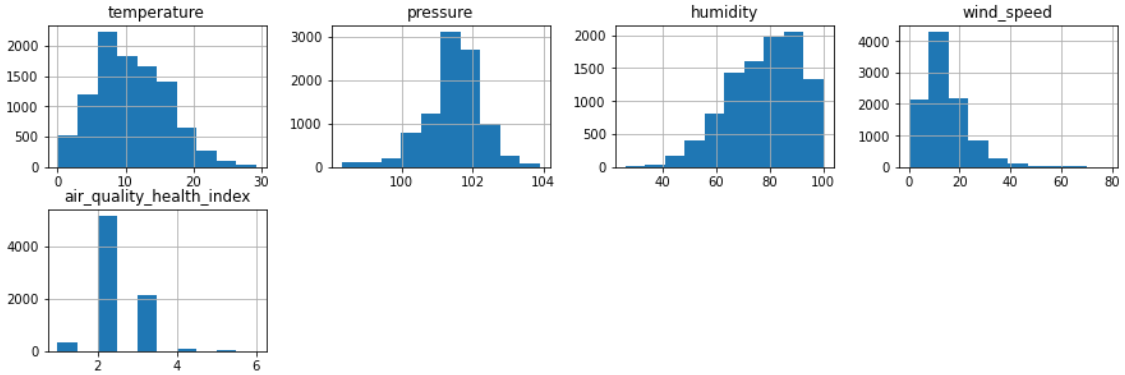


Figure 13. Histogram of entire data

B. *Density Plot*

Density plots can be used for visualizing the continuous numeric variables in the dataset. It can be used to get information about the data to check whether it follows Gaussian distribution or not[10]. The pressure and temperature variables are following a rough gaussian distribution.

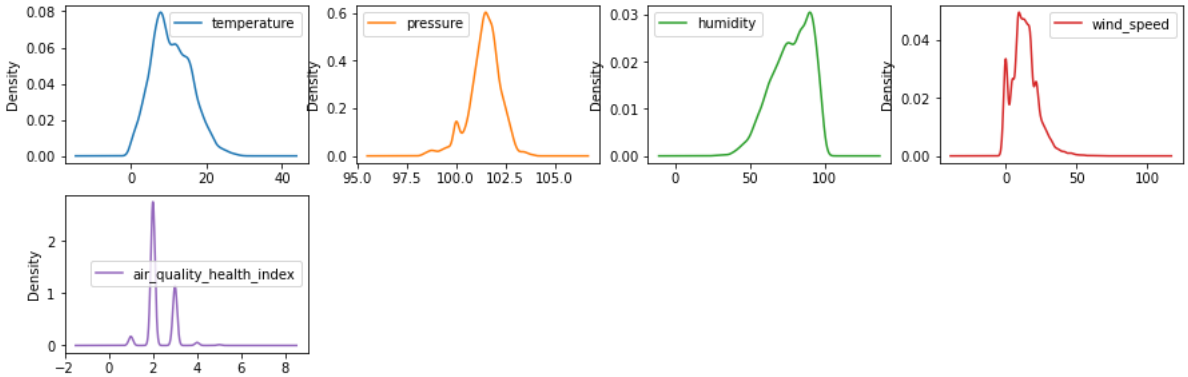


Figure 14. Density plot of entire data

C. *Box and Whisker Plots*

Box plots can visualize the continuous numerical variables. Box plot gives the information about the outliers more effectively. The interquartile range can be calculated from the box plot and also the variance, the minimum and maximum value can be analyzed from the box plot[11]. Here, all the variables are skewed and have outliers in them.

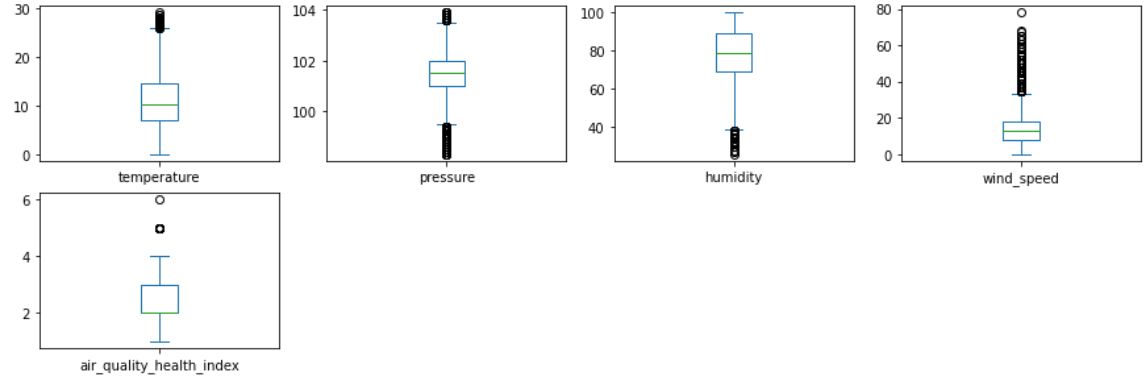


Figure 15. Box and whisker plot of entire data

D. *Correlation Matrix plot*

Correlation is a measure of how the variables are related to each other. Correlation can be directly or inversely proportion.

- The correlation value -1: Strong negative linear correlation between the two variables.

- The value 0: No linear correlation between two variables.
- The value 1: Strong positive linear correlation between two variables[12].

The variables pressure and temperature are moderately positive correlated and windspeed and temperature is less correlated. All the other variables are negatively correlated to each other.

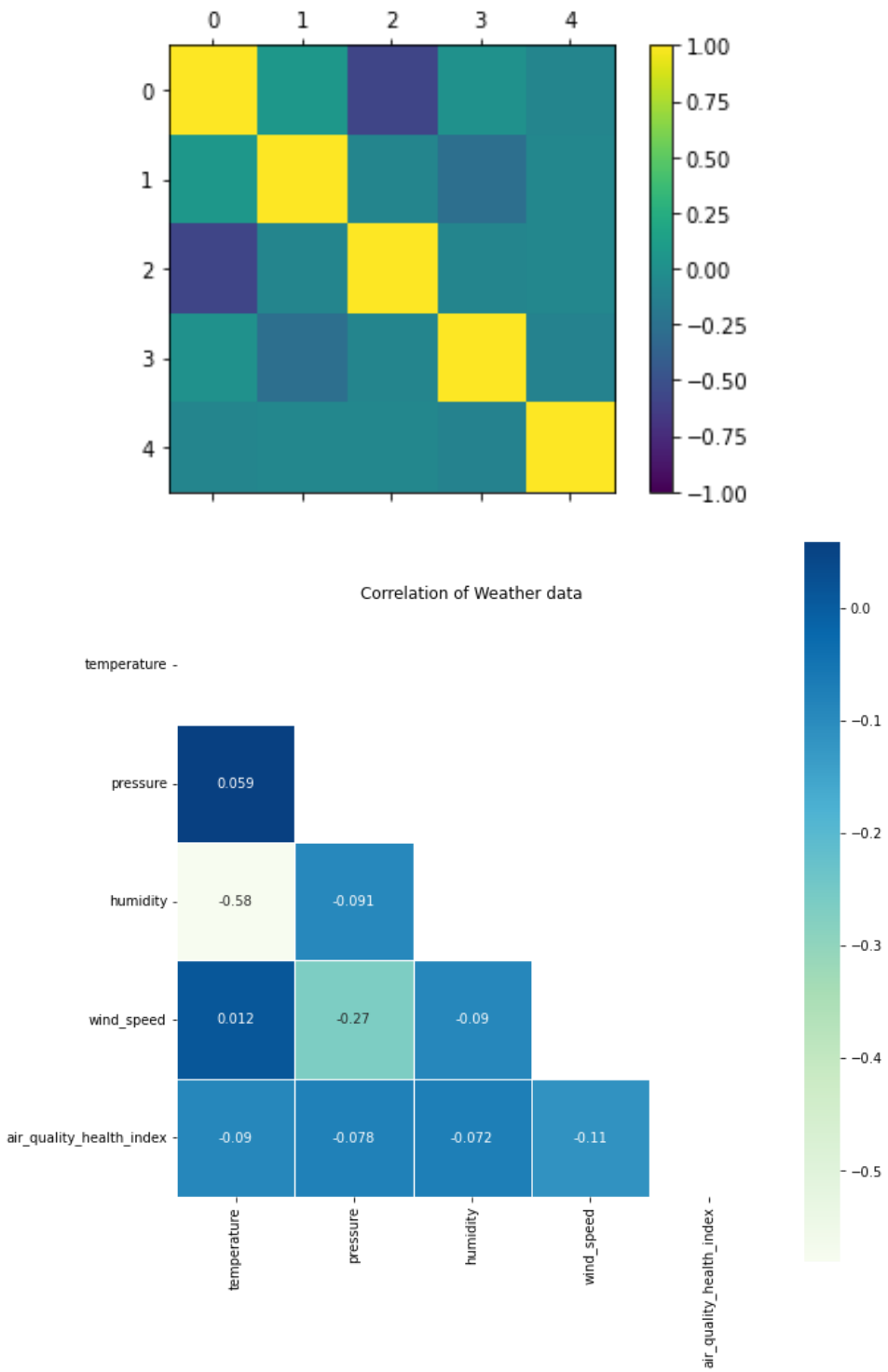


Figure 16. Correlation matrix plot of entire data

E. Univariate Visualizations

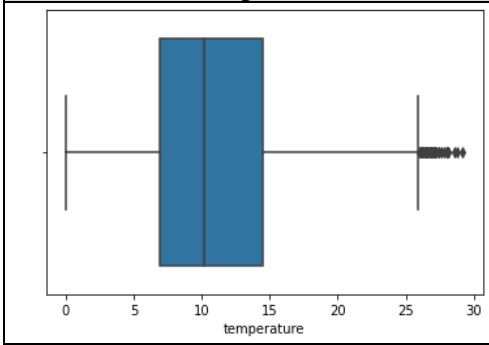
temperature

Interpretation
• There are outliers in the data and is right skewed. • The whiskers are of different length and the mean and median are different. • So, the data is not symmetrical.

TABLE III. Univariate plot for *temperature*

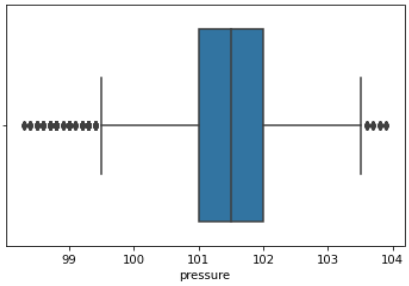
pressure

Interpretation
• There are outliers in the data and is skewed. • The length of whiskers are different and the mean and median are roughly equal. • The data is asymmetrical.

TABLE IV. Univariate plot for *pressure*

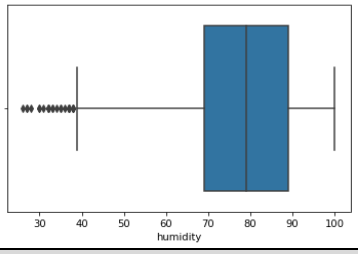
humidity

Interpretation
• The data has outliers and is left skewed. • The length of whiskers are different and the mean and median are roughly equal. • So, the data is asymmetrical.

TABLE IV. Univariate plot for *humidity*

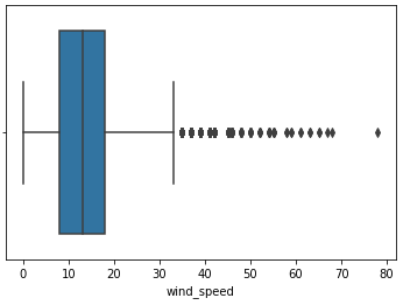
wind_speed

Interpretation
• There are outliers in the data and is right skewed. • The length of whiskers are different and the mean and median are roughly equal. • So, the data is not symmetrical.

TABLE VI. Univariate plot for *wind_speed*

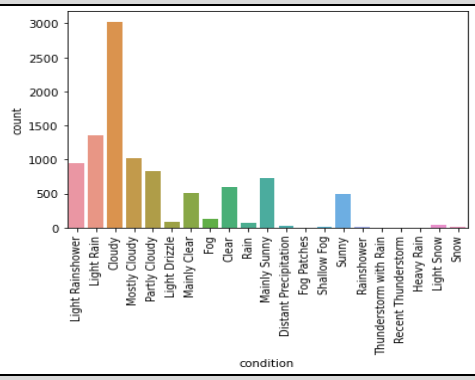
condition

Interpretation
• Most of the days were cloudy. • There were very less days with thunderstorm and heavy rain.

TABLE VII. Univariate plot for *condition*

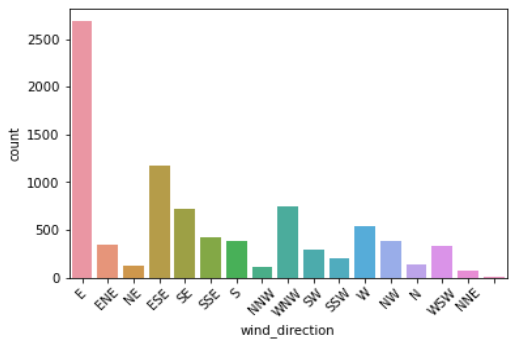
wind_direction

Interpretation
• The wind direction is East in most days and North-Northeast in least days.

TABLE VIII. Univariate plot *wind_direction*

F. Bivariate Visualizations

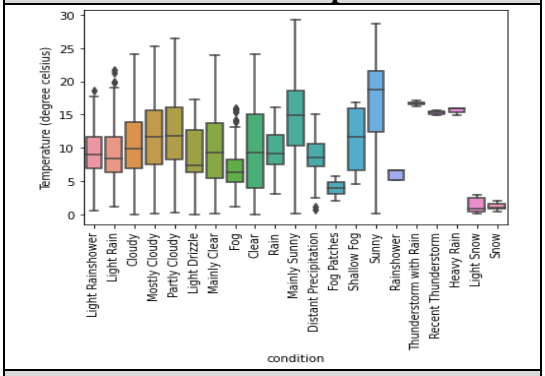
condition and temperature

Interpretation
• Sunny days have the highest median. • There are outliers present in some group.

TABLE IX. Bivariate plot for *condition* and *temperature*

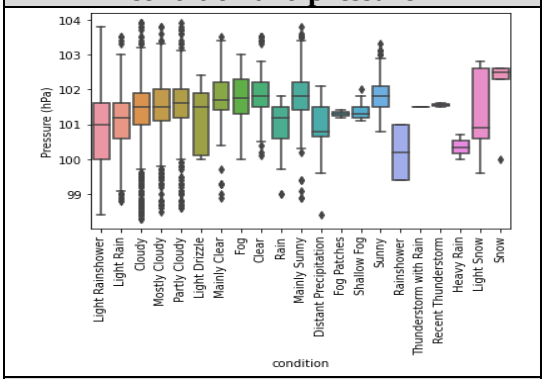
condition and pressure

Interpretation
• There are outliers present in some group.

TABLE X. Bivariate plot for *condition* and *pressure*

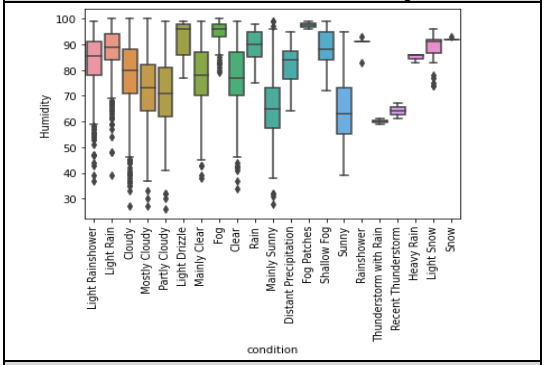
condition and humidity

Interpretation
• There are outliers present in some group.

TABLE XI. Bivariate plot for *condition* and *humidity*

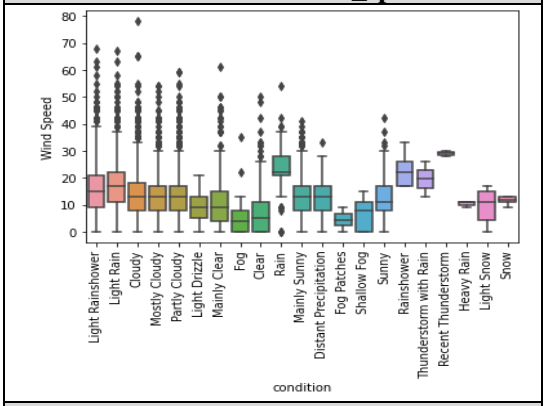
condition and wind_speed

Interpretation
• There are outliers present in some group.

TABLE XII. Bivariate plot for *condition* and *wind_speed*

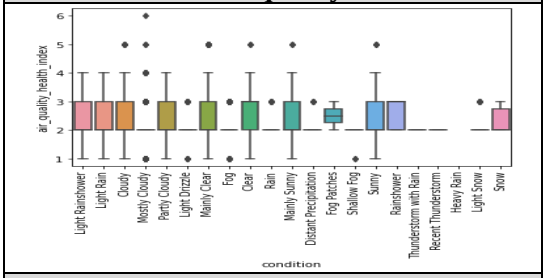
condition and air_quality_health_index

Interpretation
• The median is found to be overlapped with either the lower quartile or the upper quartile. • There are outliers present in some group.

TABLE XIII. Bivariate plot for *condition* and *air_quality_health_index*

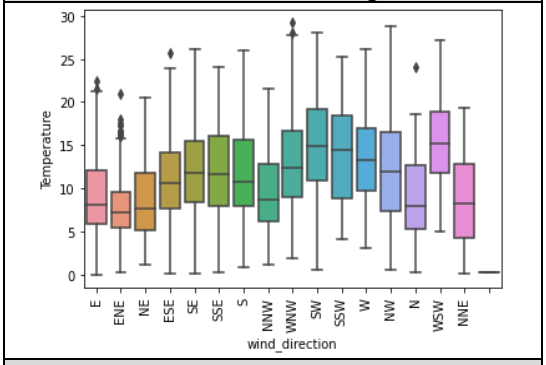
wind_direction and temperature

Interpretation
• There are outliers present in some group.

TABLE XIV. Bivariate plot for *wind_direction* and *temperature*

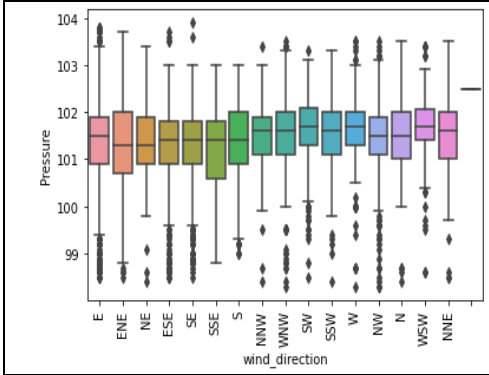
wind_direction and pressure

Interpretation
• There are outliers present in some group.

TABLE XVII. Bivariate plot for *wind_direction* and *pressure*

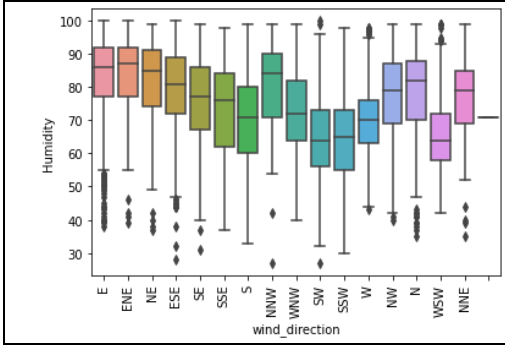
wind_direction and humidity

Interpretation
• There are outliers present in some group.

TABLE XVIII. Bivariate plot for *wind_direction* and *humidity*

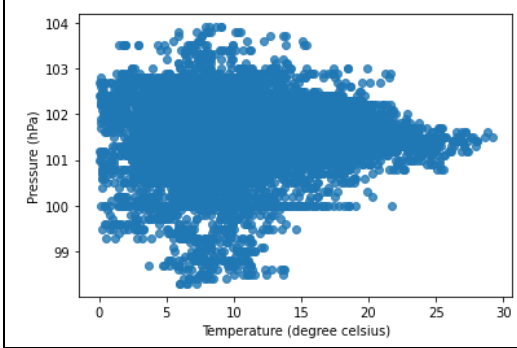
temperature and pressure

Interpretation
• There is a small positive relation with the temperature and the pressure. • The datapoints are clustered together.

TABLE XIX. Bivariate plot for *temperature* and *pressure*

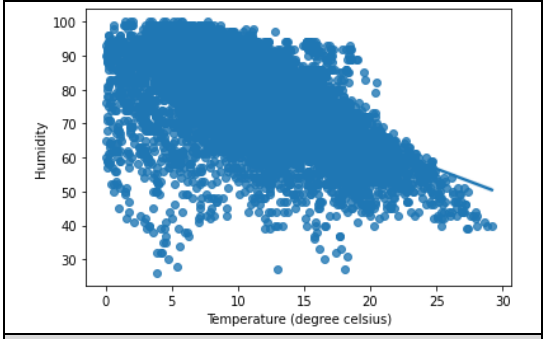
temperature and humidity

Interpretation
• There is a small negative relation with the temperature and the humidity. • The datapoints are clustered together.

TABLE XX. Bivariate plot for *temperature* and *humidity*

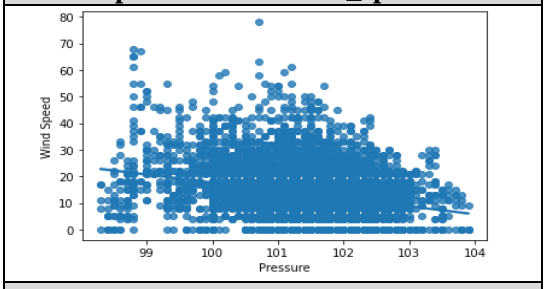
pressure and wind_speed

Interpretation
• There is no relation with the pressure and the wind speed. • The datapoints are clustered together.

TABLE XXI. Bivariate plot for *pressure* and *wind_speed*

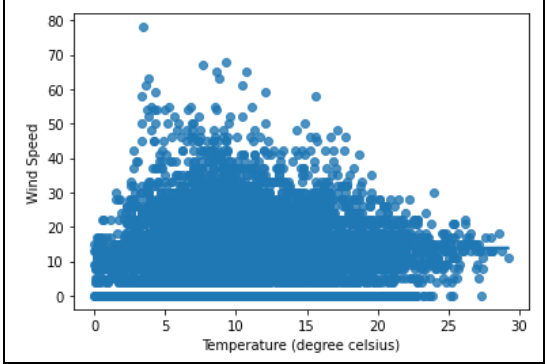
temperature and wind_speed

Interpretation
• There is no relation with the temperature and the wind speed. • The datapoints are clustered together.

TABLE XXII. Bivariate plot for *temperature* and *wind_speed*

G. *Visualizations of temperature data for time series analysis*

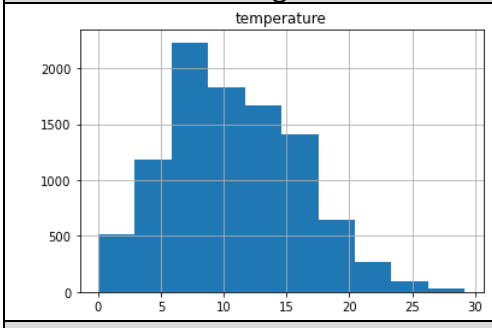
Histogram

Interpretation
• The data is roughly gaussian distributed.

TABLE XXIII. Histogram for *temperature* data

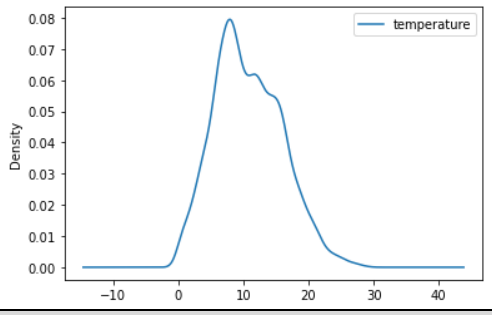
Density plot

Interpretation
• The data is roughly gaussian distributed.

TABLE XXIV. Density plot for *temperature* data

H. *Lag Plot*

A lag plot ensures whether a data is random or not. Random data should not show any specific structure in the lag plot. The non-random pattern in the lag plot indicates that the data is not random.[13]

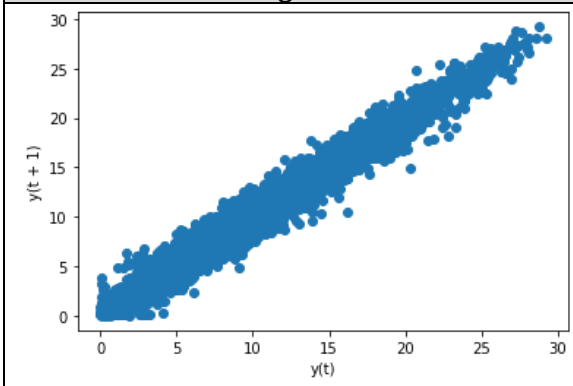
Lag Plot

Interpretation
• This lag plot reveals a linear pattern. • This indicates that the data are strongly non-random. • So, an autoregressive model might be appropriate.

TABLE XXV. Lag Plot for *temperature* data

VI. Research Questions And Methods

Analyzing the weather data helped to understand and visualize the patterns in the data and to build the time series model. The following research questions were emerged while analyzing the data:

- Which model will give the best r squared value for the available data?
- How does the window shifting affect the model?
- Which transform methods will be better for the dataset for better analysis?
- Which resampling method is suitable for the time series analysis with the available data?

The exploratory data analysis helped to understand the data distribution and the presence of the outliers in the data. Since the temperature column has no missing values, it is better to perform the time series analysis with that variable. The dataset contains the hourly temperature readings and hence it must be resampled to make a small sample. The histogram and density plot for the temperature data is found to be roughly symmetrical. The autoregressive model will be appropriate for the data based on the lag plot. The different methods used in the project are described below.

A. Resampling Methods

Resampling is a method of frequency conversion of time series data. It can be done using the resample function[14]. Resampling helps to check how data behaves differently under different frequency. There are two types of resampling.

1. Upsampling: Resampling method where the frequency of samples is increased such as from hours to minutes.
2. Downsampling: Resampling method where the frequency of the samples is reduced such as from days to months.

Both downsampling and upsampling methods were used to understand the difference in the data with both methods. The dataset was converted to another dataset with the required time, date and temperature information needed for the analysis. The data is downsampled to daily and monthly data using the resample() function. The resampled daily data is with 371 observations and the monthly data contains 14 observations. The data is then upsampled to minutes data from the hour data. The upsampled data has 17765 observations. There are NaNs present in the upsampled data since the resampling method increased the frequency of the sample.

B. Interpolation Methods

Interpolation is used to fill the NaNs created after performing resampling. Here there are NaNs in the data after upsampling. There are different interpolation techniques like pad, quadratic, linear and spline. I have used the padding interpolation technique to fill the NaNs. Interpolation through padding means copying the value just before a missing entry. We need to specify a limit while using the padding interpolation. The limit is the maximum number of NaNs the method can fill consecutively[15]. The NaNs can be also replaced by the forward and backward filling method. The bfill or backward-fill moves the first identified non-null value backward until another non-null value is found[16]. And the 'ffill' or 'forward fill' will move last valid observation forward[17].

C. Sliding Window Shifting

Sliding window is a method to add the lag features. Lag features are used to transform the time series forecasting problems into supervised learning problems[18]. The sliding window shift with multiple steps are performed to analyze the lag features.

D. Transforming Methods

Data transforms are used to remove noise and enhance the signal in time series forecasting. There are different power transform methods. It includes the square root transform, Log Transform, Transform with constant and Box Cox Transform[19]. The transform techniques were performed on the original dataset as well as the resampled data to verify the results.

By taking the square root of a time series data with a quadratic growth pattern, it can be converted to linear data. This is the square root transform. Log transform can be used to transform the time series with an exponential distribution to linear. This is done by taking the log of the values. Since the log transforms are efficient at removing exponential variance, they are widely used with time series data. The Box-Cox transform method transform the data to normal distribution. It supports both square root and log transformation, as well as a set of other related transforms[19].

E. Smoothing Methods

Smoothing is a technique used in time series to get rid of the variation between time values. Moving averages are one of the smoothing methods used in time series forecasting. A moving average needs a window size known as the window width which explains how many raw observations were used to calculate the moving average value. [20]. The Centered and Trailing Moving Average are the two main types of moving averages.

VII. Model Building And Results

The data was analyzed using different plots prior to modelling and the data was made fit for modelling using the resampling, smoothing, and transforming techniques. Downsampling was found to be appropriate for the dataset. The window shifted data can be used for machine learning models to predict the weather. From the transforming techniques it is found that the original data was transformed to a roughly gaussian distributed data after the square root transformation. But the downsampled data was not reduced to Gaussian. So, the log transform, the box cox transforms and transform with constant were applied on the original dataset. The square root transform was found to be appropriate with the available dataset. The moving averages smoothing techniques were applied on the data to make the predictions. As new observations are added daily, the model can be updated, evaluated and a prediction can be made for the next day[20]. The moving averages data was found to be closely following the actual data. Exponential smoothing is also applied to the data and optimal alpha value is found be 0.99.

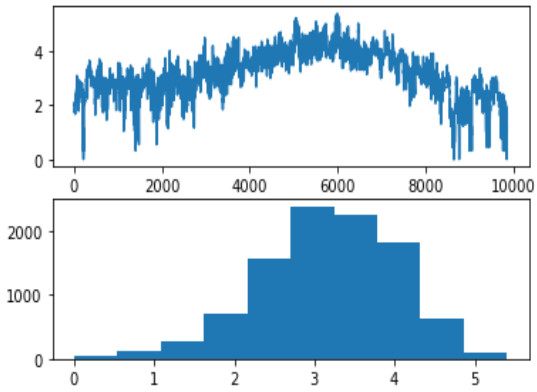


Figure 17. Square root transformed data

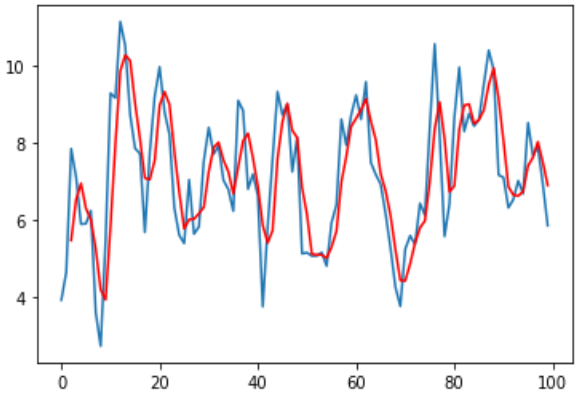


Figure 18. Moving averages

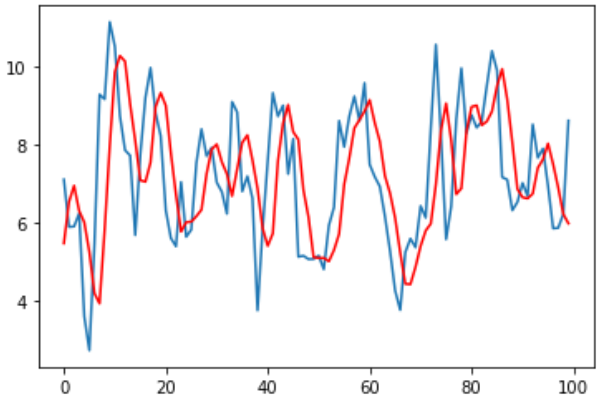


Figure 19. Moving averages for prediction

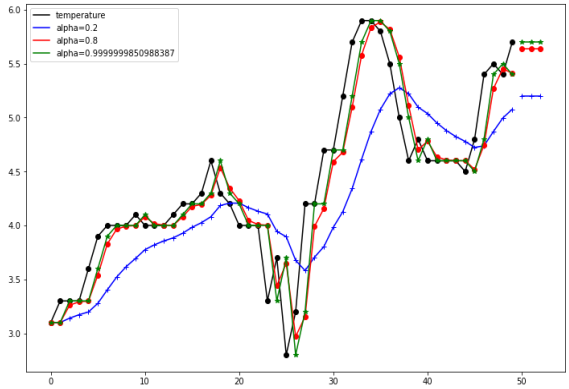


Figure 20. Exponential smoothing

A. *ARIMA Model*

Time series is a collection of data points gathered at constant time intervals. Time series analysis is used to determine the long-term trend to forecast the future. ARIMA, also known as 'Auto Regressive Integrated Moving Average' is a class of models that describes a given time series based on its own past values[21]. The first step to build the ARIMA model is to make the data stationary. Here, the resampled monthly data is used for time series analysis with ARIMA model. This is because, the original hourly data has data recorded at each hour from 8 am to 11 pm in each day. But for some dates does not have the hourly data from 8-11. Hence, it makes the frequency inconstant. So, it is appropriate to use the resampled data with constant frequency for modelling.

First the stationarity of the data is checked using rolling statistics and Augmented Dickey-Fuller Test (ADF). In the ADF test, the time series is considered stationary if the p-value is less than the 0.05 and the critical values at 1%, 5%, 10% confidence intervals are as close as possible to the ADF Statistics[22]. In the analysis, the Pvalue was found to be greater than 0.05 and so the data is nonstationary. By taking the log of the data, the data can be made stationary. The log of the data is taken and then ADF test is performed again to check the p-value. It is found to be

less than 0.05 and hence the data is stationary. The p, d, q values must be calculated to perform the model. This can be done by the auto correlation and partial autocorrelation graphs.

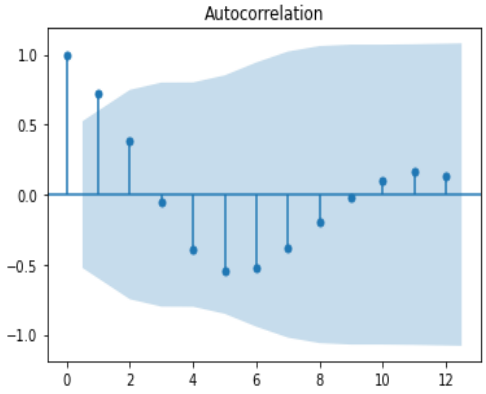


Figure 21. Autocorrelation

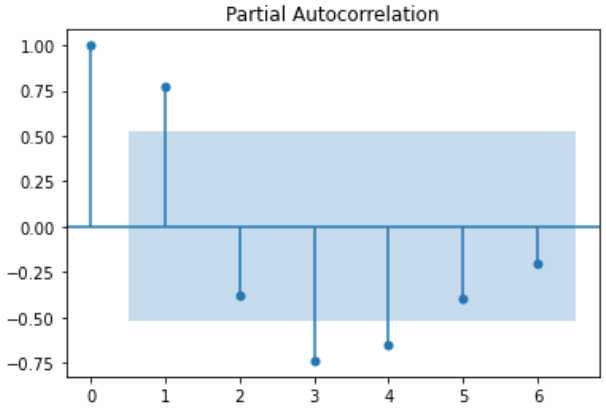


Figure 22. Partial autocorrelation

In the auto correlation and partial auto correlation graph, 2 lags are above the confidence interval and hence we can choose q and p as 2.

The grid search approach for ARIMA model is used to calculate the RMSE value to identify the best model by giving a list of p and q values. The best model was found to be of order (1, 2, 0) with p = 1, d = 2 and q =0 and RMSE value of 0.319.

ARIMA Model Results

Dep. Variable:	D2.y	No. Observations:	12
Model:	ARIMA(1, 2, 0)	Log Likelihood	-0.177
Method:	css-mle	S.D. of innovations	0.238
Date:	Mon, 02 May 2022	AIC	6.354
Time:	03:20:59	BIC	7.809
Sample:	2	HQIC	5.815

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0753	0.042	-1.800	0.072	-0.157	0.007
ar.L1.D2.y	-0.7393	0.266	-2.775	0.006	-1.262	-0.217

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-1.3526	+0.0000j	1.3526	0.5000

Figure 23. ARIMA model results

For the best model the AIC should be less as possible[23]. Here the AIC for the best model is 6.35. The BIC is 7.8 and HQIC is 5.8. The actual and forecast values are plotted using the plot_predict. The residual plot is also plotted to verify the distribution of the data.

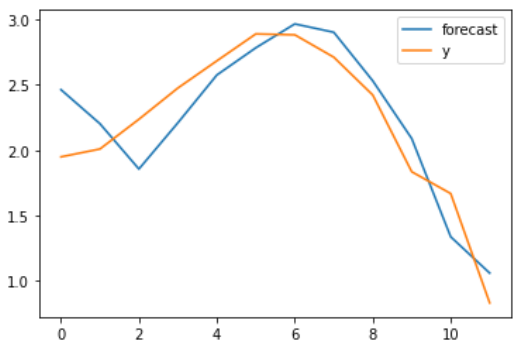


Figure 24. Forecast Vs Actual values Plot

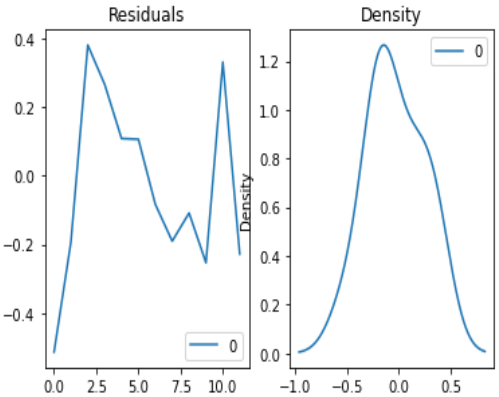


Figure 25. Residuals errors plot

The forecast values are found to be differing more at the peaks. Changes in the auto regression parameters, the moving average and differencing can affect these values. We can see, there is a very small bias in the model. Ideally, the mean should have been zero. Here the mean is -0.03 from the descriptive statistics of residual error. Since the bias is very small and the residual plot is roughly Gaussian, I am not performing the bias correction.

Auto ARIMA model was performed to identify the best fit model based on the AIC values.

```
Performing stepwise search to minimize aic
 ARIMA(0,2,0)(0,0,0)[0] intercept   : AIC=8.605, Time=0.17 sec
 ARIMA(1,2,0)(0,0,0)[0] intercept   : AIC=6.354, Time=0.08 sec
 ARIMA(0,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.13 sec
 ARIMA(0,2,0)(0,0,0)[0]             : AIC=8.311, Time=0.02 sec
 ARIMA(2,2,0)(0,0,0)[0] intercept   : AIC=8.238, Time=0.14 sec
 ARIMA(1,2,1)(0,0,0)[0] intercept   : AIC=8.241, Time=0.19 sec
 ARIMA(2,2,1)(0,0,0)[0] intercept   : AIC=inf, Time=0.30 sec
 ARIMA(1,2,0)(0,0,0)[0]             : AIC=7.304, Time=0.04 sec

Best model:  ARIMA(1,2,0)(0,0,0)[0] intercept
Total fit time: 1.089 seconds
                               SARIMAX Results
==========================================================================================
Dep. Variable:                           y   No. Observations:                   14
Model:               SARIMAX(1, 2, 0)   Log Likelihood                -0.177
Date:                Mon, 02 May 2022   AIC                           6.354
Time:                        15:33:09   BIC                           7.809
Sample:                             0   HQIC                          5.815
                                  - 14
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -0.1310      0.083     -1.572      0.116      -0.294       0.032
ar.L1         -0.7393      0.272     -2.716      0.007      -1.273      -0.206
sigma2         0.0565      0.048      1.180      0.238      -0.037       0.150
===================================================================================
Ljung-Box (L1) (Q):                   0.05   Jarque-Bera (JB):                 1.11
Prob(Q):                              0.83   Prob(JB):                         0.57
Heteroskedasticity (H):               0.64   Skew:                             0.39
Prob(H) (two-sided):                  0.67   Kurtosis:                         1.74
===================================================================================
```

Figure 26. Auto ARIMA results

The best fit model was found to be the ARIMA (1,2,0) with AIC value of 6.354.

B. Linear Regression Model

Linear regression model was implemented with the shifted data. Single step shifted data was used for this. The data was split into train and test data for this purpose and the regression model is performed. The accuracy of the model is found to be 97%. Linear regression was used because of the nature of the variables. The NaN values from the shifted data was removed to apply the model.

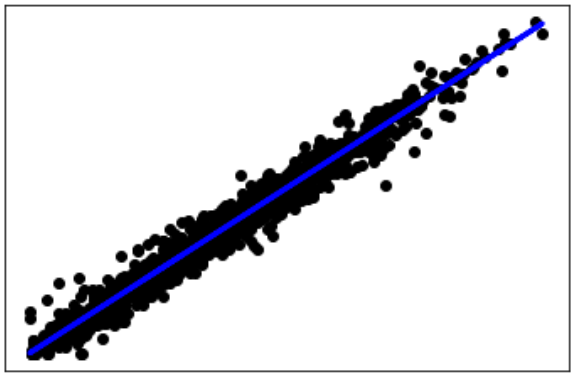


Figure 26. LR output plot

VIII. CONCLUSION

The forecasting of weather data can help the humankind in many ways. The weather forecast helps to predict the natural calamities like flood, hurricane and helps the people to be safe. The agricultural sectors, industries are depending on the weather and weather forecasting. Hence the weather forecasting helps to support the economy of the society and the quality of life. In this project, the weather data is analyzed to understand the patterns in the data and to build the time series model. Exploratory data analysis is performed to visualize the data and from that it is verified that the temperature data is roughly Gaussian distributed. With the available data, downsampling was found to be appropriate for better analysis. The ARIMA model is performed after making the data stationary because

ARIMA works well with stationary data. The best model is found to be ARIMA (1,2,0) with RMSE of .0319. Though the data cannot predict the exact temperature, it can be used to get information that helps to make strategies for proper planning of agriculture or can be used as a tool for environmental planning.

IX. Ethical Considerations

There are no ethical issues related to the project. The data for the analysis is collected from the Harvard Dataverse and it was available to the public for downloading and using. References are given to the other publications and authors wherever necessary to avoid ethical issues.

X. References

[1] G. Odero, "Time Series Analysis of Weather Data in South Carolina," 2019, Accessed: May 01, 2022. [Online]. Available: https://scholarcommons.sc.edu/etd

[2] "What Is Weather? | Center for Science Education." https://scied.ucar.edu/learning-zone/how-weather-works/weather (accessed May 01, 2022).

[3] Paras and S. Mathur, "A Feature Based Neural Network Model for Weather Forecasting ", Accessed: May 01, 2022. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.193.5077&rep=rep1&type=pdf

[4] N. Kumar and G. Kumar Jha, "A Time Series ANN Approach for Weather Forecasting," *International Journal of Control Theory and Computer Modeling (IJCTCM)*, vol. 3, no. 1, 2013, doi: 10.5121/ijctcm.2013.3102.

[5] A. K. Shukla, Y. A. Garde, and I. Jain, "Forecast of weather parameters using time series data," *Mausam*, vol. 65, no. 4, pp. 509–520, Dec. 2014, doi: 10.54302/MAUSAM.V65I4.1185.

[6] P. M. Mukadi and C. González-García, "Time Series Analysis of Climatic Variables in Peninsular Spain. Trends and Forecasting Models for Data between 20th and 21st Centuries," *Climate 2021, Vol. 9, Page 119*, vol. 9, no. 7, p. 119, Jul. 2021, doi: 10.3390/CLI9070119.

[7] J. Barzola-Monteses, M. Mite-León, M. Espinoza-Andaluz, J. Gómez-Romero, and W. Fajardo, "Time series analysis for predicting hydroelectric power production: The ecuador case," *Sustainability (Switzerland)*, vol. 11, no. 23, Dec. 2019, doi: 10.3390/SU11236539.

[8] "weather.tab - Harvard Dataverse." https://dataverse.harvard.edu/file.xhtml?persistentId=doi:10.7910/DVN/2K9FFE/NVDUAT&version=1.7 (accessed May 02, 2022).

[9] "5.7 Histogram." https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch9/histo/5214822-eng.htm (accessed May 02, 2022).

[10] "Density – from Data to Viz." https://www.data-to-viz.com/graph/density.html (accessed May 02, 2022).

[11] "A Complete Guide to Box Plots | Tutorial by Chartio." https://chartio.com/learn/charts/box-plot-complete-guide/ (accessed May 02, 2022).

[12] "Correlation Coefficients: Positive, Negative, & Zero." https://www.investopedia.com/ask/answers/032515/what-does-it-mean-if-correlation-coefficient-positive-negative-or-zero.asp (accessed May 02, 2022).

[13] "1.3.3.15. Lag Plot." https://www.itl.nist.gov/div898/handbook/eda/section3/lagplot.htm (accessed May 01, 2022).

[14] "Resample and Interpolate time series data - kanoki." https://kanoki.org/2020/04/14/resample-and-interpolate-time-series-data/ (accessed May 02, 2022).

[16] "Python | Pandas dataframe.bfill() - GeeksforGeeks." https://www.geeksforgeeks.org/python-pandas-dataframe-bfill/?ref=lbp (accessed May 02, 2022).

[17] "Python | Pandas dataframe.ffill() - GeeksforGeeks." https://www.geeksforgeeks.org/python-pandas-dataframe-ffill/ (accessed May 02, 2022).

[18] "Basic Feature Engineering With Time Series Data in Python." https://machinelearningmastery.com/basic-feature-engineering-time-series-data-python/ (accessed May 02, 2022).

[19] "How to Use Power Transforms for Time Series Forecast Data with Python." https://machinelearningmastery.com/power-transform-time-series-forecast-data-python/ (accessed May 02, 2022).

[20] "Moving Average Smoothing for Data Preparation and Time Series Forecasting in Python." https://machinelearningmastery.com/moving-average-smoothing-for-time-series-forecasting-python/ (accessed May 02, 2022).

[21] "ARIMA Model - Complete Guide to Time Series Forecasting in Python | ML+." https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/ (accessed May 02, 2022).

[22] "How to Build ARIMA Model in Python for time series forecasting?" https://www.projectpro.io/article/how-to-build-arima-model-in-python/544 (accessed May 02, 2022).