**Certificate**

Certified that **Om Singhal , Hardik Sharma , Pranav Sharma** and **Ishaan Mittal** has carried out the project work presented in this report entitled **“Weather Forecast System”** for the award of **Bachelor of Technology** from ABES Engineering College, Ghaziabad, under my supervision. The report embodies result of original work and studies carried out by Student himself/herself and the contents of the report do not form the basis for the award of any other degree to the candidate or to anybody else.

Date: Ms. Shalini Singh

# Acknowledgement

We take this opportunity to thank our teachers and friends who helped us throughout the project.

First and foremost, I would like to thank my guide for the project (**Ms Shalini Singh, ABES Engineering College , Deparment of Computer Science**) for her/his valuable advice and time during development of project.

We would also like to thank **Dr. Pankaj Kumar Sharma(HOD, Computer Science Department**) for his constant support during the development of the project.

Om Singhal Hardik Sharma

2000320120120 2000320120077

Pranav Sharma Ishaan Mittal

2000320120126 200320120086

# Declaration

*We hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.*

*Signature:*

*Name: Om Singhal(*2000320120120*)*

*Name: Hardik Sharma(*2000320120077*)*

*Name: Pranav Sharma(*2000320120126*)*

*Name: Ishaan Mittal(*2000320120086)

*Date:*

# Abstract

* Data-driven approaches, most prominently deep learning, have become powerful tools for prediction in many domains. A natural question to ask is whether data-driven methods could also be used to predict global weather patterns days in advance. First studies show promise but the lack of a common data set and evaluation metrics make intercomparison between studies difficult.
* The quantitative and reliable assessment (i. e., prediction) of the uncertainty of weather forecasts is important, both for scientific and economic reasons.
* To overcome these difficulties, the improved and reliable weather prediction methods are required. These predictions affect a nation’s economy and the lives of people.
* The motivation to achieve this outcome has led us to creating our own improvised yet simple **WEATHER FORECAST SYSTEM USING PYTHON.**

**TABLE OF CONTENTS**

**1. INTRODUCTION**

1. Problem Definition. Objective(s) of the project
2. Methodology
3. Tools and technologies used

**2. LITERATURE REVIEW**

**3.SAMPLE CODES**

**4. RESULTS/OUTPUT TESTING**

**5. CONCLUSION**

**6. REFERENCES**

**1. INTRODUCTION**

1.1 Problem Definition

1. The main ideas here is to make a high precision, accurate Weather Forecast System using machine learning techniques.
2. The frequent and drastic changes in weather has led to many difficulties around the world in the daily lives of people belonging to all the sectors of the economy.
3. This **WEATHER FORECAST SYSTEM** has been made in keeping in mind the uncertainty of weather conditions, so as to predict near to accurate weather.

1.2 Methodology

1. Data Collection: The data set used for this work is Delhi’s weather data from the year 1996 to 2017.
2. Data Pre-processing and Feature Extraction: All attributes of the data set are processed once, and the most prominent features of the data set are extracted to be used further.
3. Deriving Another Data set: From the primary data set, another data set was derived in this work.
4. Training and Testing the Machine Learning Model.
5. Data Normalization: The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without affecting differences in the ranges of values or losing any information. Normalization help in faster training of models.

Training Data

Machine Learning Algorithm

Classifier

Prediction

New Data

1.3 Tools and Techniques used

1. Pandas - Open source data structuring and analysis.
2. NumPy: Used for scientific computing in Python
3. Matplotlib: For visualization of data
4. ARIMA: Autoregressive Integrated Moving Averagewhich is a standard statistical model for time series forecast and analysis.
5. Dickey Fuller test: To check stationarity.

## 

## 2. Literature Review

Deep learning, a branch of machine learning based on multilayered artificial neural networks, has proven to be a powerful tool for a wide range of tasks, most notably image recognition and natural language processing (LeCun et al., 2015). More recently, deep learning has also been used in many fields of natural science. Much of the success of deep learning is based on the ability of neural networks to recognize patterns in high-dimensional spaces. A natural question to ask then is whether deep learning can also be used to predict future weather patterns.

Currently, weather (and climate) predictions are based on purely physical computer models, in which the governing equations, or our best approximation thereof, of the atmosphere and ocean are solved on a discrete numerical grid (Bauer et al., 2015). Overall, this approach has been very successful. However, today's numerical weather prediction (NWP) models still have shortcoming for many important applications, for example, forecasting mesoscale convective systems over Africa (Vogel et al., 2018).

ML can be applied to weather prediction in many different ways. Two long-standing applications of ML are postprocessing—the correction of statistical biases in the output of physical models—and statistical forecasting—the prediction of variables not directly output by the physical model. Traditionally, this has been done using simple linear techniques but more recently modern machine learning approaches like random forests or neural networks have been explored (Gagne et al., 2014; Lagerquist et al., 2017; McGovern et al., 2017; Rasp & Lerch, 2018; Taillardat et al., 2016). Typically, these approaches target very specific variables or locations whereas the general evolution of the atmosphere is still predicted by a physical model.

Another application that has recently been explored using ML is nowcasting, which describes the short range (up to 6 hr) prediction of precipitation by directly extrapolating radar observation without a physical model involved (Agrawal et al., 2019; Grönquist et al., 2020; Shi et al., 2015, 2017).

Global models run many times a day using different initial conditions[8] The ECMWF, for example, runs its model 51 times twice a day, incorporating new initial data to produce 1–15-day forecasts, and extends its forecasting horizon to 32 days ahead twice a week.

### Timeseries Analysis (ARIMA Model)

For prediction we are going to use one of the most popular model for time series, **Autoregressive Integrated Moving Average (ARIMA)** which is a standard statistical model for time series forecast and analysis. An ARIMA model can be understood by outlining each of its components as follows:

* **Autoregression (AR) -** refers to a model that shows a changing variable that regresses on its own lagged, or prior, values.  
  The notation **AR(p)** indicates an autoregressive model of order p.

Example — If p is 3 the predictor for X(t) will be

X(t) = µ + X(t-1) + X(t-2) + X(t-3) + εt

Where ε is error term.

* **Integrated (I) -** represents the differencing of raw observations to allow for the time series to become stationary, i.e., data values are replaced by the difference between the data values and the previous values.
* **Moving average (MA) -** incorporates the dependency between an observation and a residual error from a moving average model applied to lagged observations.

The notation **MA(q)** refers to the moving average model of order q:

X(t) = µ + εt + θ1.ε(t-1) + θ2.ε(t-2) + θ3.ε(t-3)

Here instead of difference from previous term, we take errer term (ε) obtained from the difference from past term Now we need to figure out the values of p and q which are parameters of ARIMA model. We use below two methods to figure out these values -

**Autocorrelation Function (ACF):** It just measures the correlation between two consecutive (lagged version). example at lag 4, ACF will compare series at time instance t1…t2 with series at instance t1–4…t2–4

**Partial Autocorrelation Function (PACF):**is used to measure the degree of association between X(t) and X(t-p).

## 3. SAMPLE CODE

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')

import seaborn as sns *# for plot visualization*

from statsmodels.tsa.arima\_model import ARIMA

from statsmodels.tsa.stattools import adfuller, acf, pacf

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

df = pd.read\_csv('testset.csv')

df = pd.read\_csv('testset.csv', parse\_dates=['datetime\_utc'], index\_col='datetime\_utc')

df.head()

df = df.loc[:,[' \_conds', ' \_hum', ' \_tempm']]

df = df.rename(index=str, columns={' \_conds': 'condition', ' \_hum': 'humidity', ' \_pressurem': 'pressure', ' \_tempm': 'temprature'})

print(f'dataset shape (rows, columns) - {df.shape}')

df.head()

df.index = pd.to\_datetime(df.index)

df.index

def list\_and\_visualize\_missing\_data(dataset):

    total = dataset.isnull().sum().sort\_values(ascending=False)

    percent = ((dataset.isnull().sum())/(dataset.isnull().count())).sort\_values(ascending=False)

    missing\_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])

    missing\_data = missing\_data[missing\_data.Total > 0]

    missing\_data.plot.bar(subplots=True, figsize=(16,9))

list\_and\_visualize\_missing\_data(df)

df.ffill(inplace=True)

df[df.isnull()].count()

df.describe()

df = df[df.temprature < 50]

df = df[df.humidity <= 100]

df.describe()

weather\_condition = (df.condition.value\_counts()/(df.condition.value\_counts().sum()))\*100

weather\_condition.plot.bar(figsize=(16,9))

plt.xlabel('Weather Conditions')

plt.ylabel('Percent')

df.plot(subplots=True, figsize=(20,12))

df['2015':'2017'].resample('D').fillna(method='pad').plot(subplots=True, figsize=(20,12))

train\_df = df['2000':'2015'].resample('M').mean().fillna(method='pad')

train\_df.drop(columns='humidity', axis=1, inplace=True)

test\_df = df['2015':'2017'].resample('M').mean().fillna(method='pad')

test\_df.drop(columns='humidity', axis=1, inplace=True)

def plot\_rolling\_mean\_std(ts):

    rolling\_mean = ts.rolling(12).mean()

    rolling\_std = ts.rolling(12).std()

    plt.figure(figsize=(22,10))

    plt.plot(ts, label='Actual Mean')

    plt.plot(rolling\_mean, label='Rolling Mean')

    plt.plot(rolling\_std, label = 'Rolling Std')

    plt.xlabel("Date")

    plt.ylabel("Mean Temperature")

    plt.title('Rolling Mean & Rolling Standard Deviation')

    plt.legend()

    plt.show()

def perform\_dickey\_fuller\_test(ts):

    result = adfuller(ts, autolag='AIC')

    print('Test statistic: ' , result[0])

    print('Critical Values:' ,result[4])

plot\_rolling\_mean\_std(train\_df.temprature)

perform\_dickey\_fuller\_test(train\_df.temprature)

plt.rcParams.update({'figure.figsize':(9,7), 'figure.dpi':120})

fig, axes = plt.subplots(3, 2, sharex=True)

axes[0, 0].plot(train\_df.values);

axes[0, 0].set\_title('Original Series')

plot\_acf(train\_df.values, ax=axes[0, 1])

axes[1, 0].plot(train\_df.temprature.diff().values);

axes[1, 0].set\_title('1st Order Differencing')

plot\_acf(train\_df.diff().dropna().values,ax=axes[1, 1])

axes[2, 0].plot(train\_df.temprature.diff().diff().values);

axes[2, 0].set\_title('2nd Order Differencing')

plot\_acf(train\_df.diff().diff().dropna().values,ax=axes[2, 1])

plt.xticks(rotation='vertical')

plt.show()

plt.rcParams.update({'figure.figsize':(9,3), 'figure.dpi':120})

fig, axes = plt.subplots(1, 2, sharex=True)

axes[0].plot(train\_df.diff().values); axes[0].set\_title('1st Differencing')

axes[1].set(ylim=(0,5))

plot\_pacf(train\_df.diff().dropna().values, ax=axes[1])

plt.show()

fig, axes = plt.subplots(1, 2, sharex=True)

axes[0].plot(train\_df.diff().values); axes[0].set\_title('1st Differencing')

axes[1].set(ylim=(0,1.2))

plot\_acf(train\_df.diff().dropna().values, ax=axes[1])

plt.show()

acf\_lag = acf(train\_df.diff().dropna().values, nlags=20)

pacf\_lag = pacf(train\_df.diff().dropna().values, nlags=20, method='ols')

plt.figure(figsize=(22,10))

plt.subplot(121)

plt.plot(acf\_lag)

plt.axhline(y=0,linestyle='--',color='silver')

plt.axhline(y=-1.96/np.sqrt(len(train\_df.diff().values)),linestyle='--',color='silver')

plt.axhline(y=1.96/np.sqrt(len(train\_df.diff().values)),linestyle='--',color='silver')

plt.title("Autocorrelation Function")

plt.subplot(122)

plt.plot(pacf\_lag)

plt.axhline(y=0,linestyle='--',color='silver')

plt.axhline(y=-1.96/np.sqrt(len(train\_df.diff().values)),linestyle='--',color='silver')

plt.axhline(y=1.96/np.sqrt(len(train\_df.diff().values)),linestyle='--',color='silver')

plt.title("Partial Autocorrelation Function")

plt.tight\_layout()

model = ARIMA(train\_df.values, order=(2,0,2))

model\_fit = model.fit(disp=0)

print(model\_fit.summary())

residuals = pd.DataFrame(model\_fit.resid)

fig, ax = plt.subplots(1,2)

residuals.plot(title="Residuals", ax=ax[0])

residuals.plot(kind='kde', title='Density', ax=ax[1])

plt.show()

model\_fit.plot\_predict(dynamic=False)

plt.show()

fc, se, conf = model\_fit.forecast(16, alpha=0.05)

fc\_series = pd.Series(fc, index=test\_df.index)

lower\_series = pd.Series(conf[:, 0], index=test\_df.index)

upper\_series = pd.Series(conf[:, 1], index=test\_df.index)

plt.figure(figsize=(12,5), dpi=100)

plt.plot(train\_df, label='training')

plt.plot(test\_df, label='actual')

plt.plot(fc\_series, label='forecast')

plt.fill\_between(lower\_series.index, lower\_series, upper\_series,

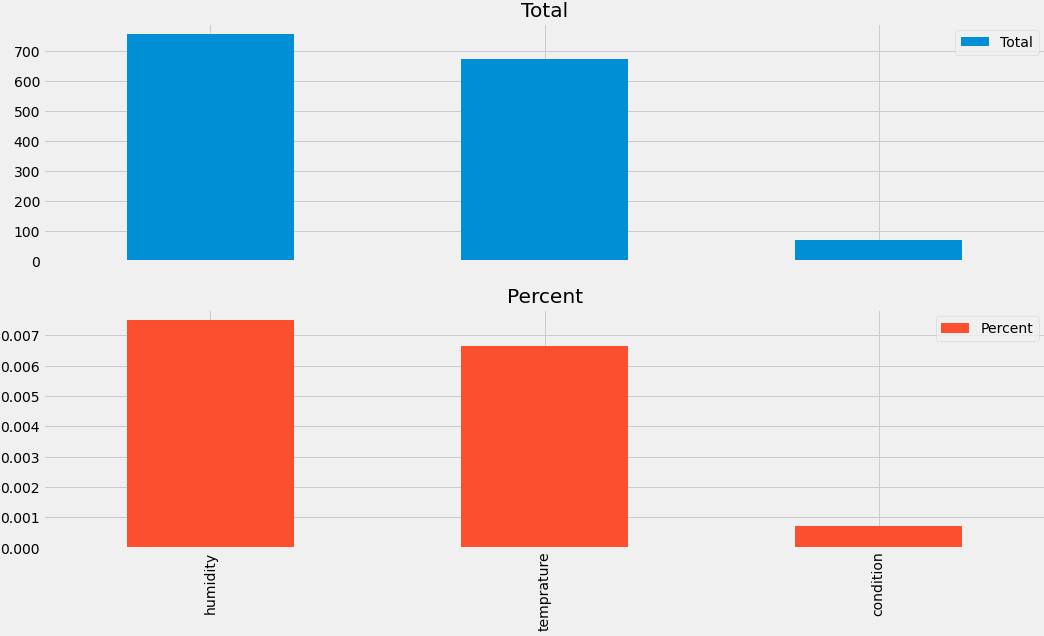
                 color='k', alpha=.15)

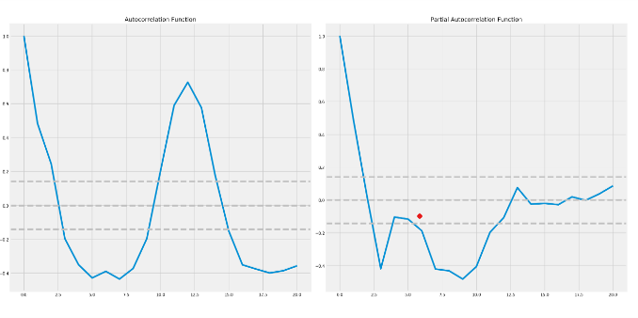
plt.title('Forecast vs Actuals')

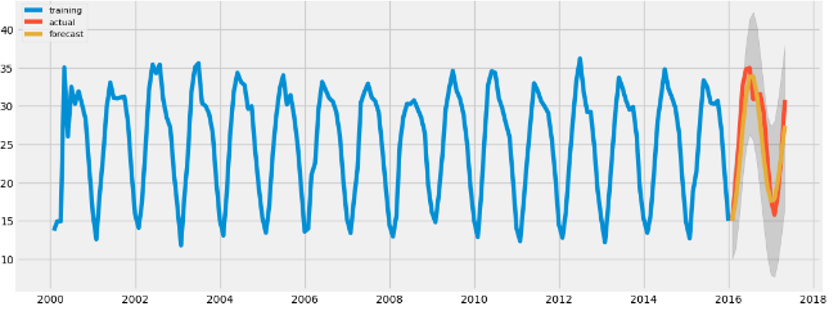
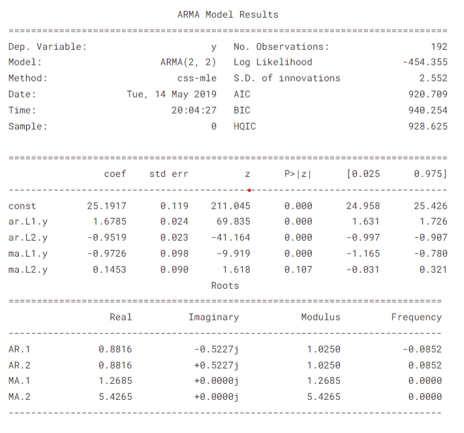
plt.legend(loc='upper left', fontsize=8)

plt.show()

**4. RESULTS/OUTPUT TESTING**







## 5. Conclusion

## 

Weather forecasting estimating is the usage of science and development to anticipate the states of the environment for a given area and time. Peoples have attempted to anticipate the climate casually for a significant long time and officially since the nineteenth century. Climate guesses are made by social occasion quantitative data about the current status of the environment at a given spot and utilizing meteorology to project how the climate will change.

Weather data is considered with different attributes for weather forecasting. The use of time series model has helped us creating a forecast model that is highly accurate with the actual model used.

This paper thus offers a survey of weather forecasting using various techniques. Also summarizing the key concepts and focusing on the existing work on weather forecasting, its types, and its applications. To conclude, how deep learning, data mining, and machine learning algorithms were employed in weather forecasting is exceptionally important to guarantee future exploration will focus destined for success, accordingly improving the performance of weather predictions. We presented a weather forecasting model that makes predictions via considerations of the joint influence of key weather variables.

## 6.References

[1] N. Hasan, M. T. Uddin, and N. K. Chowdhury, “Automated weather event analysis with machine learning,” in Proc. IEEE 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), 2016, pp. 1-5.

[2] L. L. Lai, H. Braun, Q. P. Zhang, Q. Wu, Y. N. Ma, W. C. Sun, and L. Yang, “Intelligent weather forecast,” in Proc. IEEE 2004 International Conference on Machine Learning and Cybernetics, 2004, pp. 4216-4221.

[3]Imran Maqsood, Ajith Muhammad Riaz Khan, Abraham

An ensemble of neural network for weather forecasting

Neural Comput & Applic, volume 13, p. 112 - 122Posted: 2004

[4] Delhi Weather Data. [Online]. Available: https://www.kaggle.com/mahirkukreja/delhi-weatherdata/home

[5]A. G. Salman, B. Kanigoro, and Y. Heryadi, “Weather forecasting using deep learning techniques,” in Proc. IEEE 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), 2015, pp. 281-285.

[6]Siddiqui Khalid J. and Nugen Steve M., Knowledge Based System for Weather Information Processing and Forecas-ting, Department of Computer Science, SUNY at Fredonia, NY14063, IEEE 1966.

[7]H. Saima, J. Jaafar, S. Belhaouari and T. A. Jillani, "Intelligent methods for weather forecasting: A review", 2011 National Postgraduate Conference, pp. 1-6, 2011.

[8]C Wei-Ta, H Kai-Chia and B Ali, "Visual Weather Temperature Prediction", 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 234-241, 2018

[9] Krasnopolsky, Vladimir M., and Michael S. FoxRabinovitz. ”Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction.”Neural Networks19.2 (2006): 122-134.