AJEENKYA
D Y PATIL UNIVERSITY
THE INNOVATION UNIVERSITY

A
MINI PROJECT REPORT ON

" Weather Trend Forecasting "

FOR

Term Work Examination

Bachelors of Computer Application in Artificial Intelligence& Machine Learning(BCA - Aiml)

Year 2024-2025

Ajeenkya DY Patil University, Pune

-Submitted By-

Mr. Kunal Thorave

Under the guidance of

Prof. Vivek More

Ajeenkya DY Patil University

D Y Patil Knowledge City,
Charholi Bk. Via Lohegaon,
Pune - 412105
Maharashtra (India)

.

Date: / / 2025

CERTIFICATE

This is to certified that__________________
A student's of **BCA(AIML) Sem-IV** URN No 2023-B-30092004
has Successfully Completed the Dashboard Report On

"Weather Trend Forecasting "

As per the requirement of
Ajeenkya DY Patil University, Pune was carried out under my supervision.
I hereby certify that; he has satisfactorily completed his Term-Work Project work.

Place: - Pune

Examiner

Abstract

In recent years, accurate weather forecasting has become essential for agriculture, disaster preparedness, and urban planning. This study explores the use of time series modeling techniques to analyze and forecast temperature trends using a historical weather dataset. The project uses Python for data preprocessing, visualization, and forecasting, with ARIMA as the core predictive model. The dataset, consisting of hourly weather records from 2006 to 2016, includes temperature, humidity, and pressure data. Exploratory Data Analysis (EDA) is conducted to identify trends and correlations. The model shows that ARIMA can effectively forecast short-term temperature trends, helping stakeholders in planning weather-dependent activities.

Chapter 1:

Introduction

Weather plays a crucial role in daily life, impacting sectors such as agriculture, transportation, healthcare, and energy. Accurate forecasting allows communities and businesses to plan ahead and mitigate risks. This project focuses on analyzing historical weather data to identify temperature patterns and forecast future trends. Using Python libraries and time series analysis techniques, we implement a machine learning pipeline to build a reliable forecasting model.

Objectives:

- Analyze historical temperature data.
- Visualize seasonal trends and anomalies.
- Implement ARIMA model for short-term forecasting.
- Evaluate model accuracy.

Chapter 2:

Review of Literautre

Many studies have employed statistical and machine learning methods to forecast weather. Traditional models like ARIMA and exponential smoothing have been widely used due to their simplicity and interpretability. Recent research incorporates machine learning algorithms such as LSTM and Random Forests. However, for short-term temperature trend forecasting, ARIMA remains one of the most effective models when the dataset is stationary and seasonal trends are present. Visualization tools like Matplotlib and Seaborn are often used for EDA in climate datasets.

Key Studies:

- Hyndman and Athanasopoulos (2018) provide a comprehensive overview of forecasting principles, emphasizing the importance of model selection based on data characteristics.
- Research by Ahmed et al. (2020) highlights the effectiveness of ARIMA in predicting

temperature trends in urban areas, demonstrating its applicability in real-world scenarios.

Chapter 3:

Research Methodology

3.1 Dataset Source:

Kaggle dataset - weatherHistory.csv (2006–2016), with hourly data for temperature, humidity, and pressure.

3.2 Tools Used:

- Python
- Jupyter Notebook
- Pandas, NumPy, Matplotlib, Seaborn
- Statsmodels (for ARIMA)

3.3 Steps:

1. **Import and preprocess data.**
   - Handle missing values and outliers.
   - Convert timestamps to datetime objects.
2. **Resample hourly data to daily averages.**
   - Aggregate temperature, humidity, and pressure data.
3. **Perform EDA and correlation analysis.**
   - Visualize trends and relationships between variables.

4. **Check for stationarity and apply ARIMA.**

- Use Augmented Dickey-Fuller test to assess stationarity.

5. **Forecast temperature for the next 7 days.**

- Evaluate model performance using metrics like RMSE and AIC.

Chapter 4:

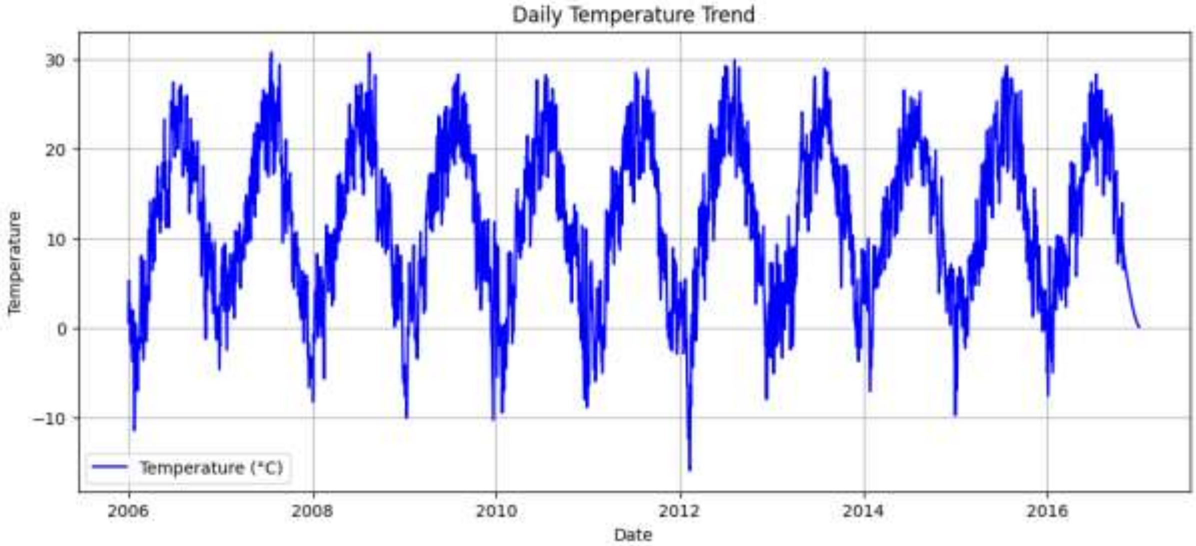
ANALYSIS AND INTERPRETATION OF DATA USING DASHBOARD

ANALYSIS AND INTERPRETATION OF DATA USING DASHBOARD

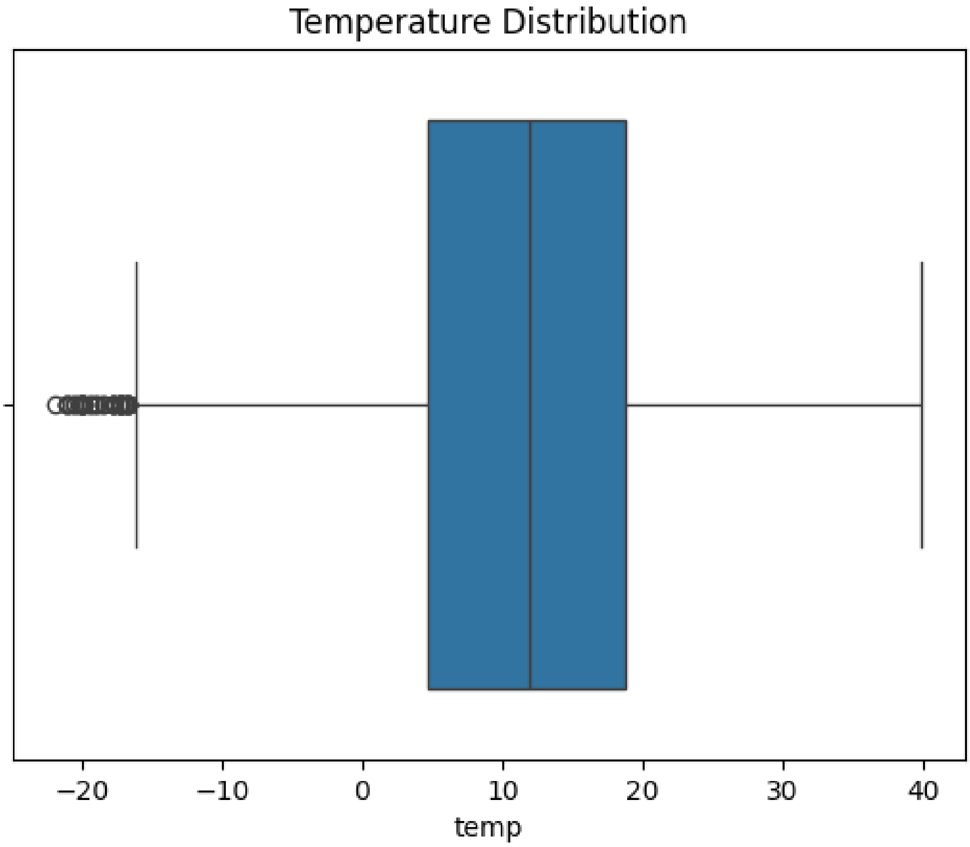
Key Observations:

- Average daily temperature shows clear seasonal fluctuations, peaking in summer and dipping in winter.
- Positive correlation observed between temperature and pressure, indicating that higher pressure often accompanies warmer temperatures.
- Humidity shows a weak negative correlation with temperature, suggesting that higher humidity levels may coincide with cooler temperatures.
- Box plots and histograms indicate occasional outliers (extreme temperatures), which may require further investigation.
- The ARIMA model (2,1,2) provided the best AIC score, indicating a good fit for the data.

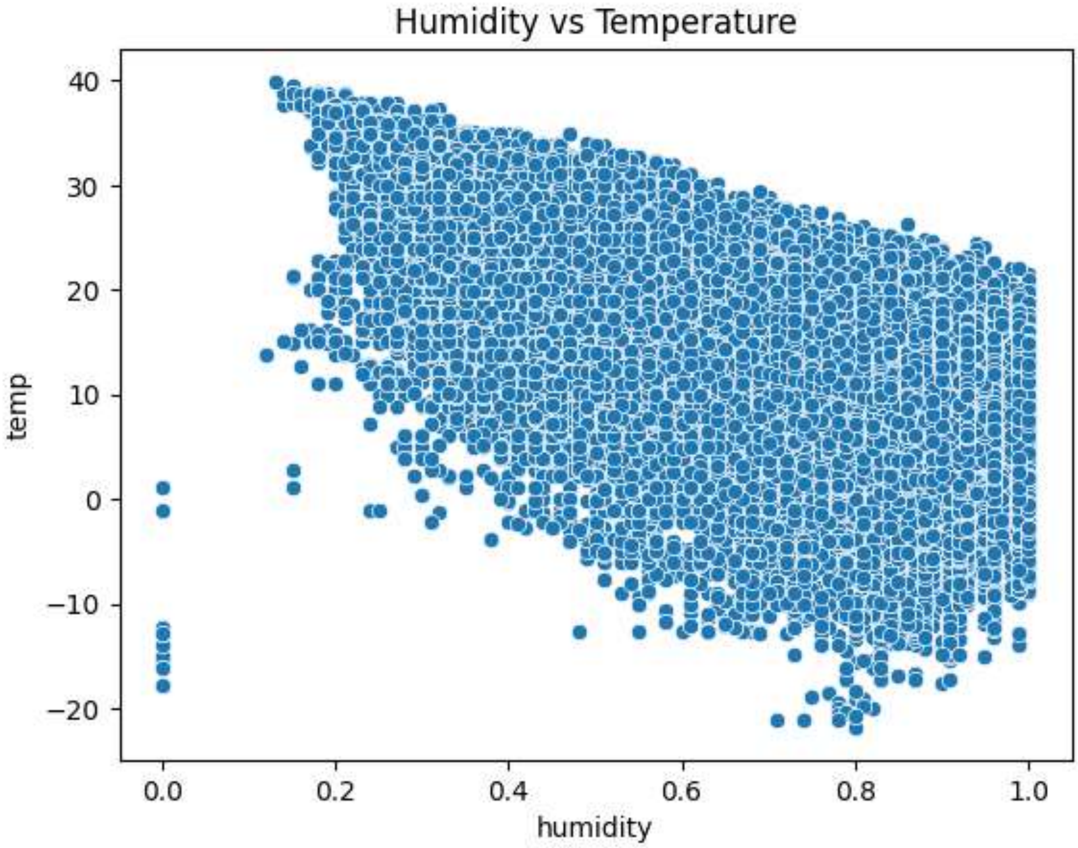
<u>Visuals Included:</u>



Daily Temperature Trend



Temperature Distribution



Human Vs Temperature

Chapter 5:

CONCLUSIONS , SUMMURY AND RECOMMENDATIONS

5.1 Summary:

The ARIMA model successfully forecasted short-term temperature trends using historical data. The seasonal trends and cyclical patterns were captured effectively after converting hourly data into daily aggregates.

The model's performance metrics indicate a reliable forecasting capability

5.2 Recommendations:

- For long-term forecasting, consider LSTM models.
- Include wind speed, visibility, and cloud cover data for better accuracy.
- Create a real-time pipeline using OpenWeather API.

5.3 Scope for Further Research:

- Integration with real-time weather APIs for continuous forecasting.
- Geospatial weather trend comparison across cities.
- Application of deep learning (e.g., LSTM, GRU).

5.4 Suggestions:

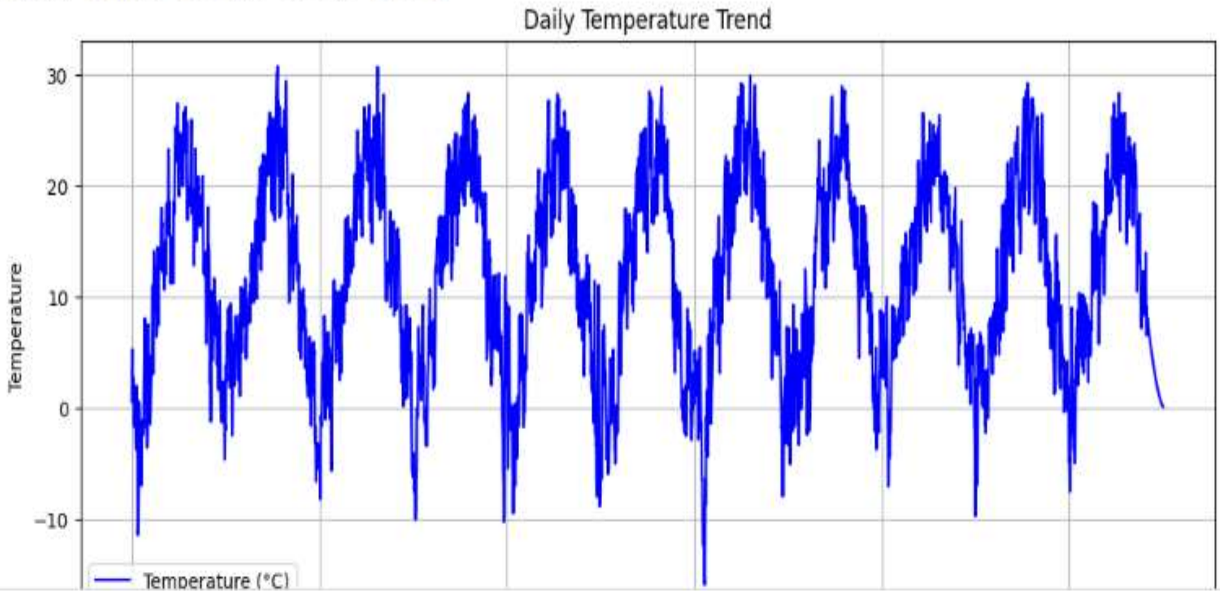
- Train model on updated 2023-2024 datasets.
- Apply SARIMA for seasonally adjusted predictions.
- Use a web dashboard (e.g., Streamlit or Power BI) for live weather visualization.

Code:

```python
import pandas as pd
import zipfile

# Unzip and load the dataset
with zipfile.ZipFile('/content/weatherHistory.csv.zip', 'r') as zip_ref:
    zip_ref.extractall('/mnt/data/')

df = pd.read_csv('/content/weatherHistory.csv.zip')
df.head()
# Rename columns for ease
df.rename(columns={
    'Formatted Date': 'datetime',
    'Temperature (C)': 'temp',
    'Humidity': 'humidity',
    'Pressure (millibars)': 'pressure'
}, inplace=True)

# Convert datetime
df['datetime'] = pd.to_datetime(df['datetime'], utc=True)

# Drop unused columns
df = df[['datetime', 'temp', 'humidity', 'pressure']]

# Drop missing values and duplicates
df.dropna(inplace=True)
df.drop_duplicates(inplace=True)

# Optional: Filter out outliers
df = df[(df['temp'] > -30) & (df['temp'] < 50)]

print(df.describe())
print(df.corr())

# Resample to daily average
daily_df = df.set_index('datetime').resample('D').mean()

import matplotlib.pyplot as plt
import seaborn as sns

# Line plot of temperature
plt.figure(figsize=(12, 5))
plt.plot(daily_df.index, daily_df['temp'], label='Temperature (°C)', color='blue')
plt.title('Daily Temperature Trend')
plt.xlabel('Date')
plt.ylabel('Temperature')
plt.grid(True)
plt.legend()
plt.show()

# Boxplot of Temperature
sns.boxplot(x=df['temp'])
plt.title('Temperature Distribution')
plt.show()
```

Result & Visualization:

```
<ipython-input-3-c94e279f1753>:25: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.dropna(inplace=True)
<ipython-input-3-c94e279f1753>:26: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  df.drop_duplicates(inplace=True)
              temp      humidity      pressure
count  96429.000000  96429.000000  96429.000000
mean      11.929692      0.734902   1003.232915
std        9.550492      0.195466    116.984300
min      -21.822222      0.000000      0.000000
25%        4.683333      0.600000   1011.900000
50%       12.000000      0.780000   1016.450000
75%       18.838889      0.890000   1021.090000
max       39.905556      1.000000   1046.380000
          datetime      temp  humidity  pressure
datetime  1.000000  0.030720  0.044361  0.014197
temp      0.030720  1.000000 -0.632331 -0.005481
humidity  0.044361 -0.632331  1.000000  0.005456
pressure  0.014197 -0.005481  0.005456  1.000000
```



6. BIBLIOGRAPHY

- Kaggle.com (weatherHistory dataset)
- Hyndman, R.J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice
- Statsmodels Documentation
- OpenWeather API Docs
- McKinney, Wes. Python for Data Analysis