

# Weather Temperature Prediction Project Report

## 1. Introduction

Weather prediction plays an essential role in human life, agriculture, industry, transportation, energy planning, and disaster management. Temperature forecasting, in particular, helps in everyday decision-making as well as long-term planning. This project focuses on **predicting minimum (tmin), maximum (tmax), and average (tavg) temperatures** using a machine-learning approach based on historical weather data from 2018 to 2024.

The project demonstrates complete data handling, cleaning, preprocessing, machine-learning model training, evaluation, and deployment through a saved model file.

---

## 2. Purpose of the Project

The main purpose of this project is to **develop an accurate temperature prediction model** using machine-learning techniques. The goals include: - Predicting real-time temperature for specific cities. - Understanding how different weather features influence temperature. - Creating a fully automated forecasting system using historical patterns. - Building practical skills in data analysis, feature engineering, and model selection.

---

## 3. Reasons for Choosing This Project

This project was selected due to: - High demand for accurate temperature forecasts. - The opportunity to work with real-world meteorological data. - The importance of weather analysis across industries. - The ability to explore multiple machine-learning methods. - A strong educational benefit in understanding data science workflows.

---

## 4. Dataset Description

The dataset used contains **7671 rows** and **11 columns** from multiple cities—Islamabad, Lahore, and Quetta—from **2018 to 2024**.

### 4.1 Dataset Columns

- **day, month, year** – Date components
- **humidity** – Atmospheric humidity percentage
- **wspd** – Wind speed
- **pressure** – Atmospheric pressure

- **dew\_point** – Dew point temperature
- **city** – Location name
- **tmin, tmax, tavg** – Minimum, maximum, and average daily temperatures

## 4.2 Missing Values Analysis

The dataset was checked thoroughly using: - `df.isnull().sum()` - `np.sum(pd.isnull(df))`

All columns contained **0 missing values**, making the dataset complete and highly reliable.

---

## 5. Data Cleaning and Preprocessing

Although the dataset had no missing values, additional cleaning steps were applied:

### 5.1 Handling City Column (Categorical Data)

The 'city' column contained text values (Islamabad, Lahore, Quetta), which machine-learning models cannot use directly.

We applied: - **Mode imputation** for safety: `df["city"].fillna(df["city"].mode()[0])` - **One-Hot Encoding** to convert city names into numerical columns: - city\_Islamabad - city\_Lahore - city\_Quetta

### 5.2 Final Dataset Shape

After encoding, the dataset expanded to **13 columns**, all numerical or boolean.

---

## 6. Feature Selection

### 6.1 Input Features (X)

```
['day', 'month', 'year', 'humidity', 'wspd', 'pressure', 'dew_point', 'city_Islamabad', 'city_Lahore', 'ci
```

### 6.2 Target Features (Y)

```
['tmin', 'tmax', 'tavg']
```

These were selected because they represent the core temperatures we want to predict.

---

## 7. Splitting the Dataset

The dataset was split into training and testing sets using:

```
train_test_split(X, Y, test_size=0.2, random_state=42)
```

- **80%** used for training - **20%** used for testing

This ensures fair evaluation of model performance.

---

## 8. Machine Learning Models Used

Three multi-output regression models were tested:

1. **RandomForestRegressor**
2. **GradientBoostingRegressor**
3. **SVR (Support Vector Regressor)**

Each model was wrapped in `MultiOutputRegressor` to allow prediction of all three temperature values simultaneously.

---

## 9. Model Training and Evaluation

Each model was: - Trained using `model.fit(X_train, y_train)` - Tested on unseen data - Evaluated using **R<sup>2</sup> score**

### 9.1 Model Scores Obtained

- **RandomForest:** 0.97914
- **GradientBoosting:** 0.96635
- **SVR:** 0.04062

### 9.2 Best Model

The **RandomForest model** performed the best with **97.9% accuracy**.

This makes it highly reliable for temperature forecasting.

---

## 10. Saving the Best Model

The final selected model was saved using:

```
joblib.dump(best_model, "best_weather_model.pkl")
```

This allows future temperature predictions without re-training the model.

---

## 11. Uses of the Project

This temperature prediction system can be used in: - **Agriculture:** Crop planning, frost protection, irrigation timing. - **Energy Management:** Predicting heating/cooling needs. - **Travel and Transportation:** Route planning and safety. - **Disaster Management:** Heat-wave and cold-wave warnings. - **Climate Research:** Analyzing long-term weather trends. - **Smart Cities:** Integration with IoT sensors.

---

## 12. Problems Solved by the Project

This project helps address: - Sudden weather uncertainty - Poor agricultural planning due to unexpected temperature shifts - Inefficient energy demand forecasting - Lack of localized, customized weather predictions - Difficulty accessing raw meteorological insights

---

## 13. Advantages of This Project Over Normal Weather Apps

- Uses **local dataset**, more specific to the selected cities
  - Provides **explainable** and **transparent** results
  - Fully customizable for research and system integration
  - Can be expanded to predict humidity, wind speed, rainfall, etc.
  - Built from scratch, providing complete learning experience
- 

## 14. Conclusion

This project successfully demonstrates how machine-learning techniques can be applied to weather forecasting. By using historical data, the model accurately predicts minimum, maximum, and average temperatures. The **RandomForest model** proved to be the most accurate and reliable.

The project showcases practical skills in: - Data preprocessing - Feature engineering - Model selection - Multi-output regression - Model saving and deployment

It provides a strong foundation for advanced weather prediction systems and real-world applications.

---

## 15. Future Scope

This project can be expanded further by: - Adding rainfall and humidity prediction - Using deep-learning models like LSTMs - Building a mobile or web app for live predictions - Integrating IoT weather sensors for real-time forecasting - Predicting extreme conditions (heatwaves, storms)

---

**End of Report**