CSE303: Statistics for Data Science
[SPRING 2024]

Term Project Report

Submitted by:

Student ID	Student Name	Contribution Percentage
2022-1-60-159	Redown Ahmed	30%
2022-1-60-131	Ankon Nandi	20%
2022-1-60-266	Tasnuva Tasnim Nova	30%
2022-1-60-065	Taniz Fatema Jarin	20%

1. Introduction

In this project, we have undertaken a series of analytical tasks using two comprehensive weather datasets for the cities, Dhaka and Delhi. Our objective is to conduct a thorough analysis and build predictive models based on these datasets. The project is divided into three key components:

- Comprehensive Exploratory Data Analysis (EDA):
  We have performed a detailed exploratory data analysis on the provided datasets. This involves summarizing the data, identifying patterns, outliers, and missing values, and visualizing various aspects of the data to gain a understanding of the distributions.

- Developing Regression Models
  We have developed regression models to predict two critical weather parameters: temperature and rainfall. The goal is to accurately forecast these parameters for both Dhaka and Delhi using the available features in the datasets. This involves selecting appropriate regression techniques, training the models and evaluating their performance.

- Developing Classification Models
  We have created classification models using Logistic Regression and Support Vector Machine (SVM) to predict the city (Dhaka or Delhi) given the input weather features. This classification distinguishes between the two cities based on their weather patterns.

2. Dataset Characteristics and Exploratory Data Analysis

In this section, introduce your dataset. Mention the number of rows, columns, and other characteristics. Provide the histograms of data distribution and correlations among the variable with a suitable discussion. Try to stand out and be creative in your presentation!

Dataset Overview: The combined dataset from Dhaka and Delhi comprises:

- **Number of Rows:** 3654 (1827 for each dataset)
- **Number of Columns:** 33

Types of Data:

- **Numerical:** tempmax, tempmin, temp, feelslikemax, feelslikemin, feelslike, dew, humidity, precip, precipprob, precipcover, snow, snowdepth, windgust, windspeed, winddir, sealevelpressure, cloudcover, visibility, solarradiation, solarenergy, uvindex, severerisk, moonphase.
- **Categorical:** name, datetime, preciptype, conditions, description, icon, stations, sunrise, sunset.

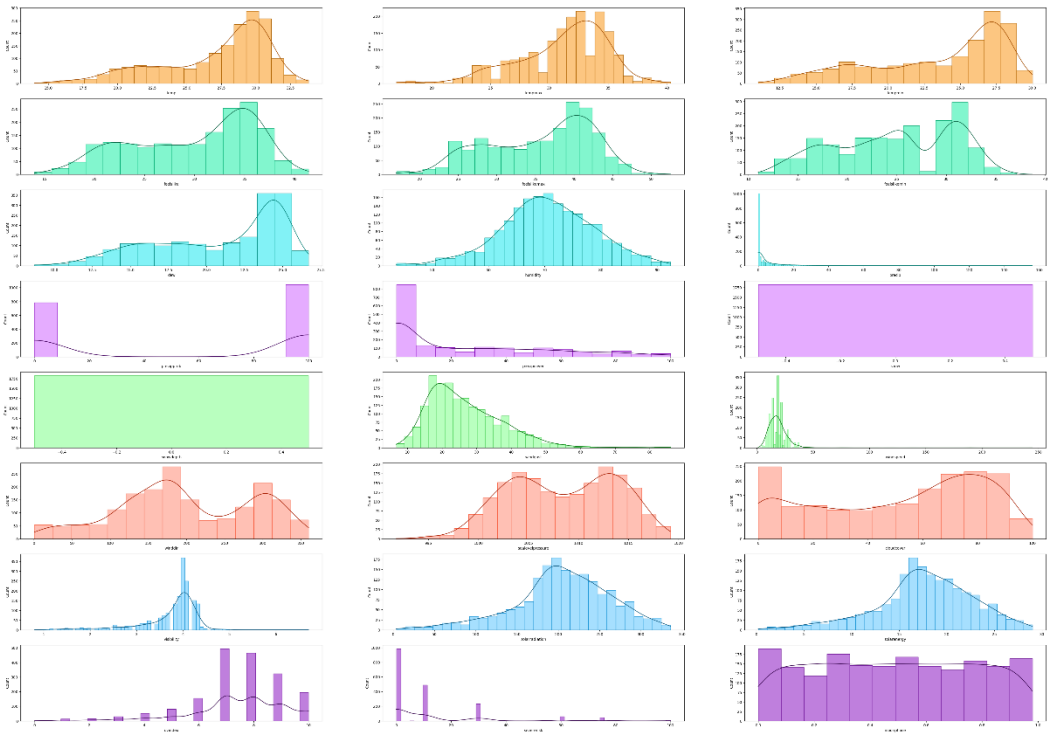


Fig: Dhaka Histplot

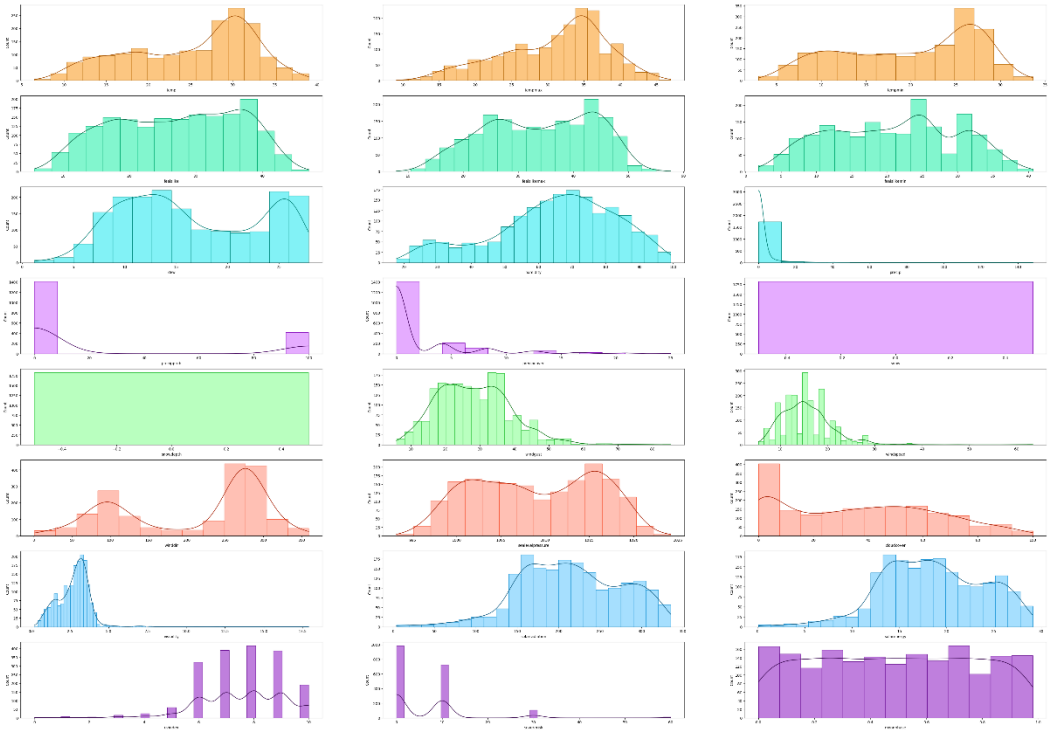


Fig: Delhi Histplot

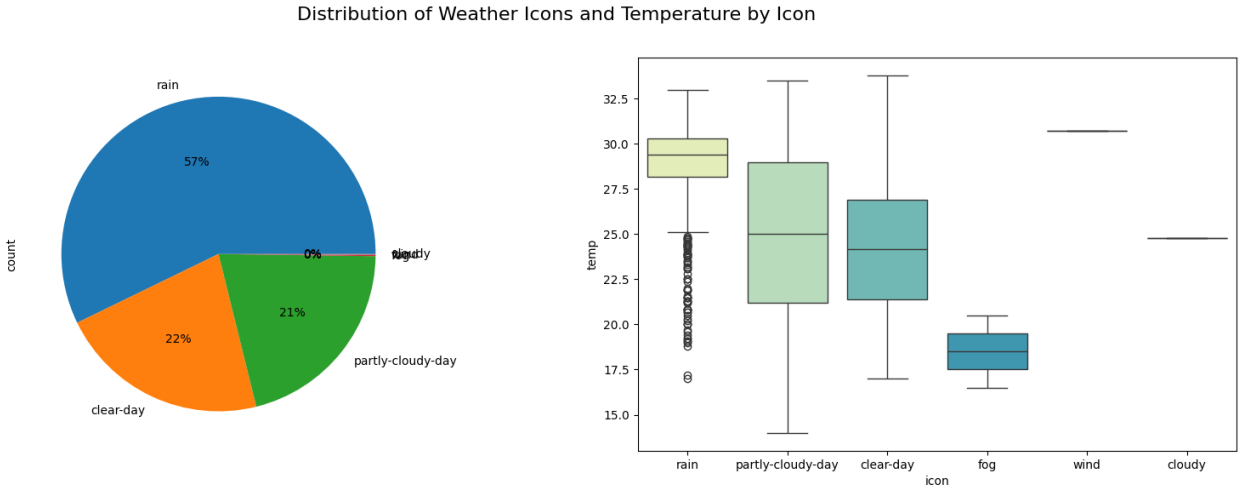


Fig: Dhaka Subplot

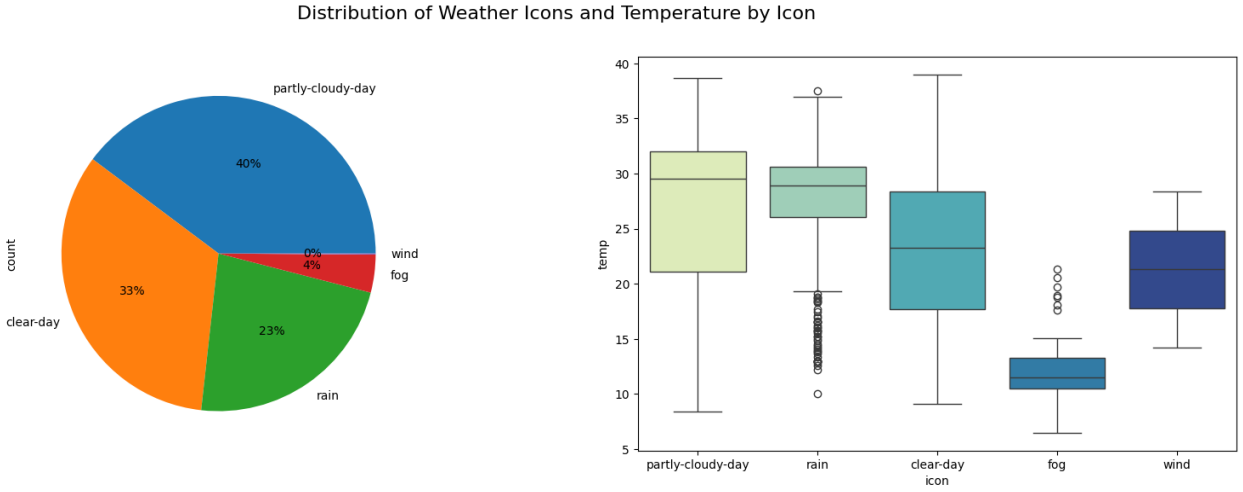


Fig: Delhi Subplot

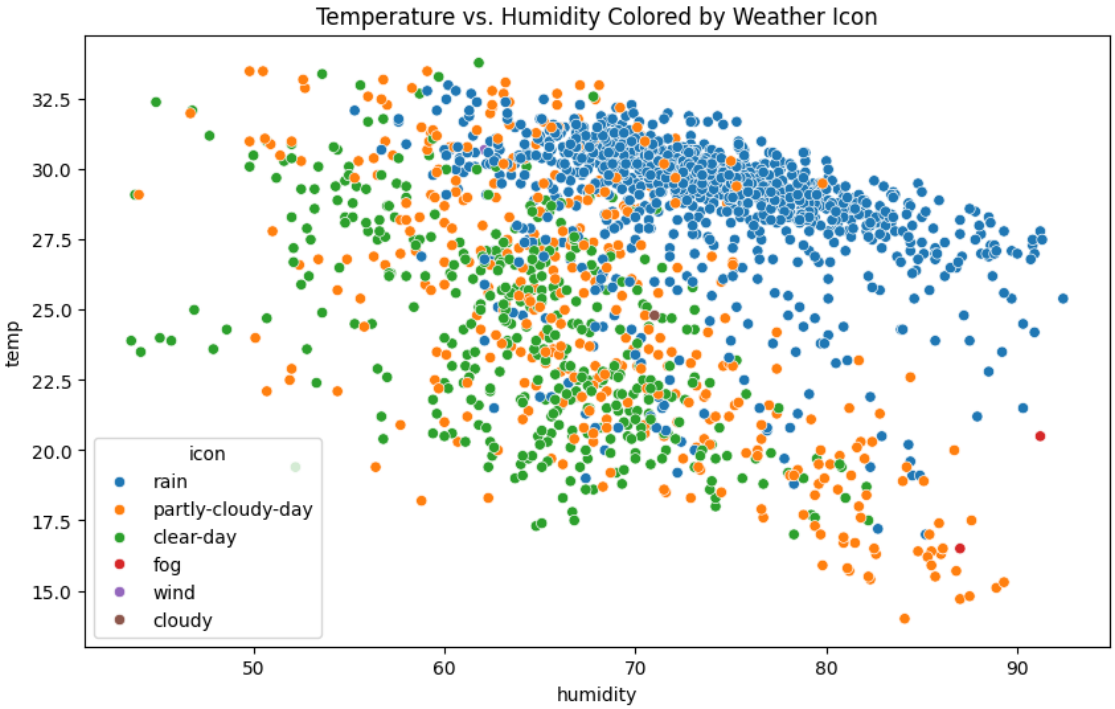


Fig: Dhaka Scatterplot

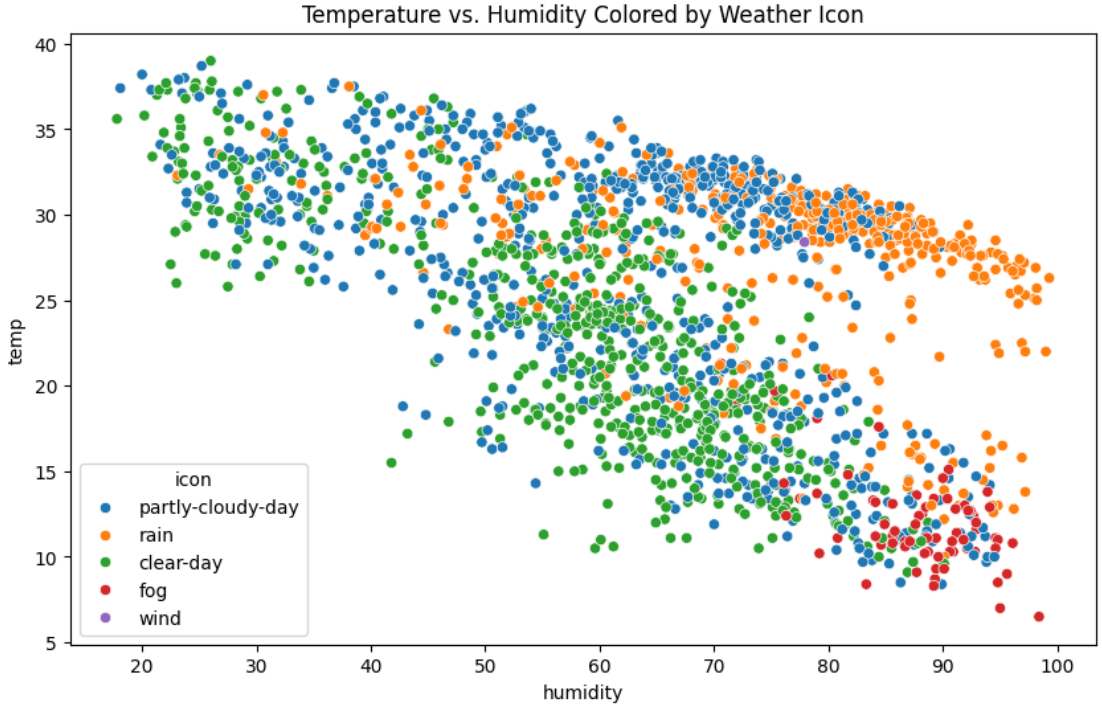


Fig: Delhi Scatterplot

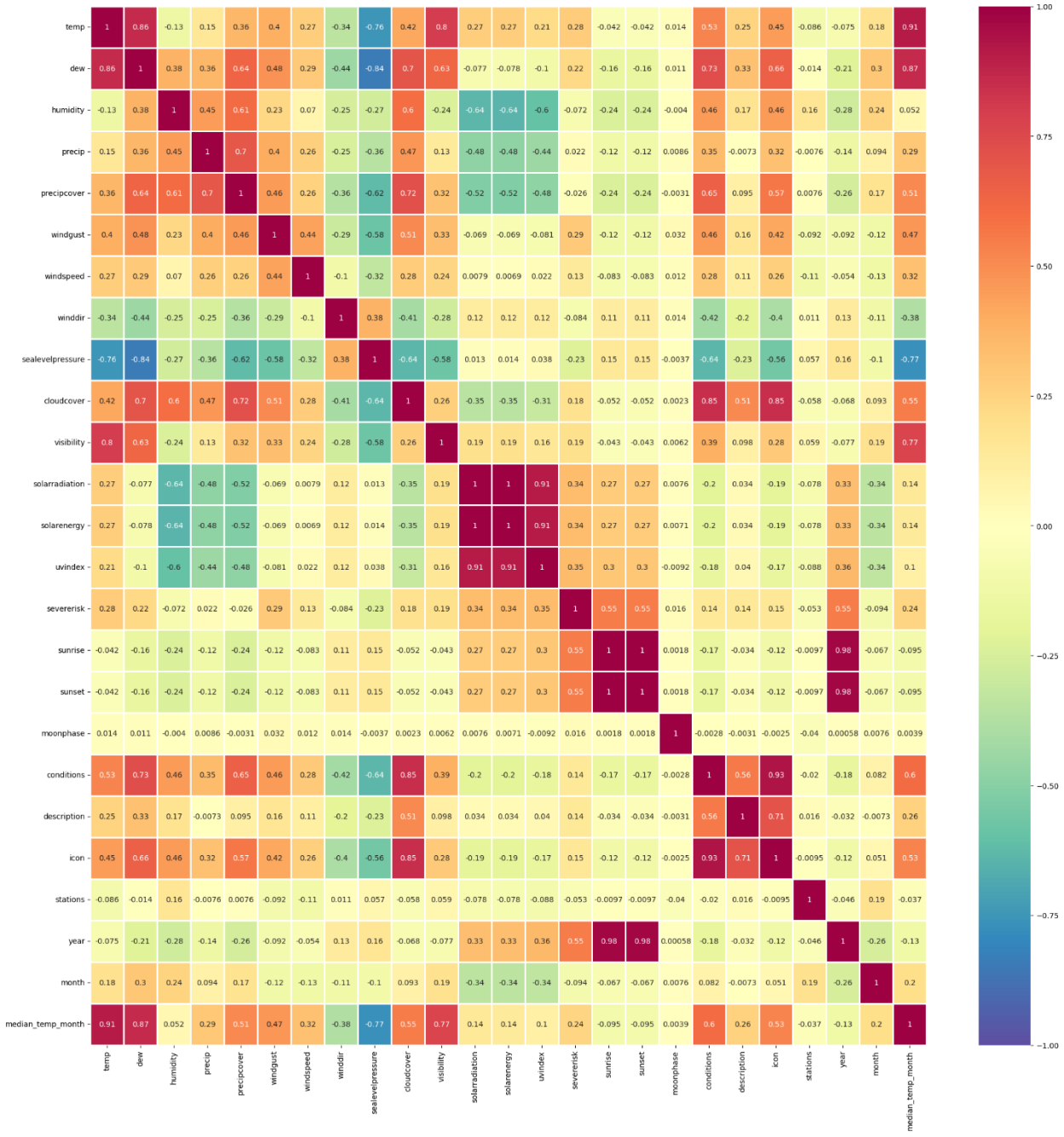


Fig: Dhaka Heatmap

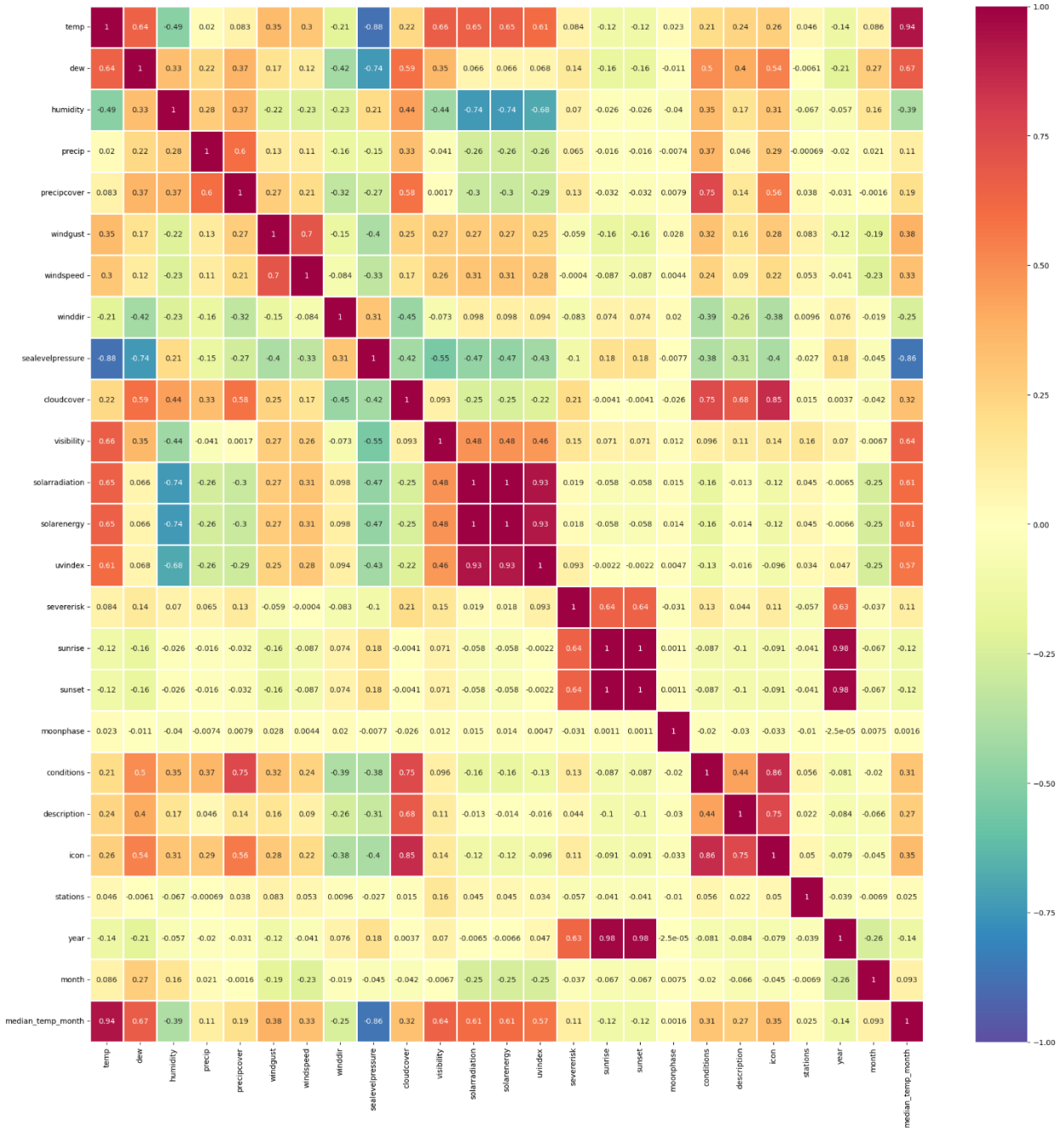


Fig: Delhi Heatmap

3. Machine Learning Models

Linear Regression

A linear regression model expresses the relationship between a dependent variable and one or more independent variables. A linear relationship between the dependent and independent variables is the underlying assumption of linear regression. The model can be written as follows:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n ,$$

where y is the dependent variable, x_1, x_2 , …, x_n are the independent variables, β_0 is the intercept, β_1 , β_2 , …, …, β_n are the coefficients. The best-fit line, is determined by the model by minimizing the sum of the squared differences, or residuals, between the dependent variable's predicted values and observed values.

Logistic Regression:

Logistic regression is a popularly used classification technique that provides a model that features the probability of binary outcomes based on one or more predictor variables. The relationship between the independent variables and the likelihood of a particular event occurring is modelled using logistic regression. It uses the logistic function, sometimes referred to as the sigmoid function, to determine the likelihood that an instance belongs to a specific class. Overfitting is another risk associated with logistic regression, especially in cases where the model has a large number of predictor variables. By penalizing large coefficient values, regularization approaches like L1 (Lasso) or L2 (Ridge) regularization can be used to prevent overfitting in logistic regression.

Support Vector Machine (SVM):

Support Vector Machine (SVM) is used for classification as well as regression tasks. By increasing the margin—the distance between the hyperplane and the closest data points from each class—SVM seeks to identify the hyperplane that best divides the classes (support vectors). The hyperplane that maximizes this margin is the ideal one. Support Vectors are the critical data points that lie closest to the decision boundary. If the data cannot be separated linearly, SVM can use a kernel function to transfer it into a higher-dimensional space, where the algorithm can identify a linear separating hyperplane. SVM is capable of handling both linear and non-linear data through the use of kernel functions. The trade-off between maximizing the margin and decreasing the classification error is managed by the regularization parameter (C). While a larger C value aims for fewer misclassifications with a narrower margin, a lesser C value permits a bigger margin with more misclassifications.

4. Data Preprocessing

- Filling Null values:
  Example:
  df['preciptype'] = df['preciptype'].fillna("not rain")
  df['preciptype'].value_counts()

  The code fills any missing values in the preciptype column of a DataFrame with the string "not rain" and then counts the occurrences of each unique value in the preciptype column. Similarly, for column severerisk .
- Extracting year and month from datetime column:
  Example:
  df['year'] = df['datetime'].dt.year
  df['month'] = df['datetime'].dt.month

  It extracts the year and month from the datetime column in our DataFrame and assign it to new columns called ‘year’ and ‘month’.

- For ‘temp’ , We have created a new column df_dhk_temp['median_temp_month']

  df_dhk_temp['median_temp_month'] =
  df['month'].map(df.groupby('month')['temp'].median())

  This code actually stores the median temp data for each month through mapping.

  Similarly, we used this technic for precip for both Dhaka and Delhi Dataset.

- Dropping columns:
  Example for Dhaka and Delhi datasets:
  drop_columns = correlation[abs(correlation) < 0.4].index.tolist()
  drop_columns
  df_temporary = df['month']
  df_dhk_temp = df_dhk_temp.drop(columns = drop_columns)
  df_dhk_temp['month'] = df_temporary
  drop_columns

  This cleans the DataFrame df_dhk_temp by removing columns that have weak correlations (absolute correlation value less than 0.4) with other variables. However, the month column is preserved and restored after the columns are dropped, ensuring that it remains in the DataFrame.

  Example for a merged dataset:
  selector = SequentialFeatureSelector(model, n_features_to_select = 4,
  scoring='accuracy')
  selector.fit(x, y)
  selected_features = selector.get_support()

print('The selected features are:', list(x.columns[selected_features]))

This efficiently selects the most relevant features for the given model using sequential feature selection. This method helps in reducing the dimensionality of the dataset while maintaining or improving the model's performance. Every other column is dropped except the columns selected by SequentialFeatureSelector.

- **Label Encoding:**
  Example:
  df_dhk_temp['sunrise'] = LabelEncoder().fit_transform(df_dhk_temp['sunrise'])
  By using LabelEncoder, the sunrise times are converted into numerical labels. This transformation is essential for machine learning algorithms, which typically require numerical input. The unique values in the sunrise column are encoded as integers, enabling the model to process the data effectively.
  Similarly, for columns: Sunset, icon, conditions, stations, description

5. Different Models

1. Linear Regression:

- Correlation by Taking the Correlation >= 0.4 and Correlation <= -0.4:
  Example:
  correlation = df_dhk_temp.corr()['temp']

2. Logistic Regression

Logistic Regression is used for predicting continuous outcomes. In this project, it was applied to predict temperature and rainfall.

- **Sequential Feature Selector:**
  selector = SequentialFeatureSelector(model, n_features_to_select = 4, scoring='accuracy')
  selector.fit(x, y)
  selected_features = selector.get_support()

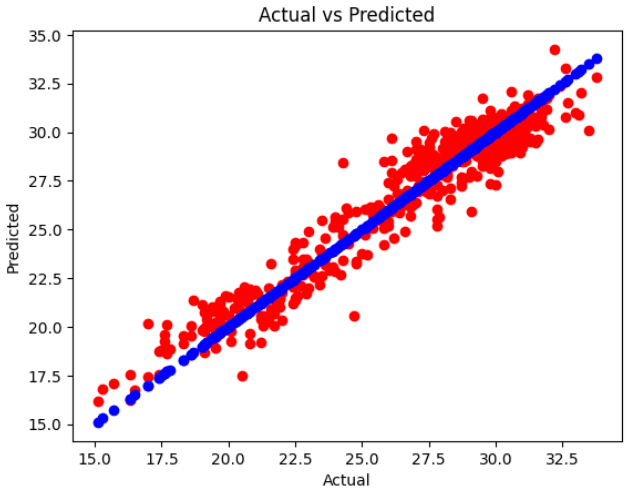
Parameter Description:

Parameter	Description
model	Logistic regression model
n_features_to_select	The number of feature/column
scoring	Features selected based on accuracy

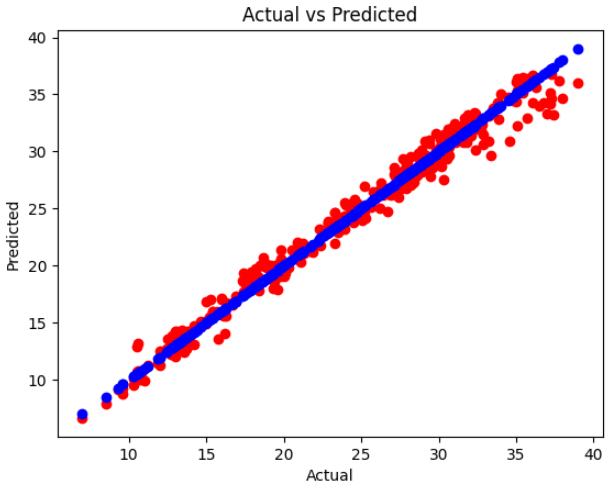
3. Support Vector Machine (SVM):
Same features and parameters mentioned above in Logistic Regression.

6. Performance Evaluation

- Actual vs Predict:
  For column ‘temp’, both datasets:

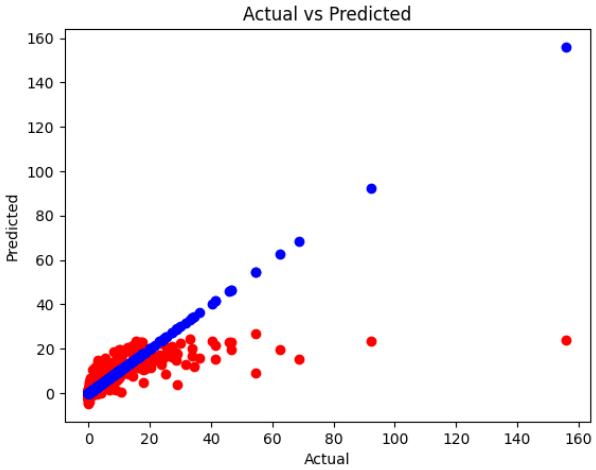


Dhaka Dataset

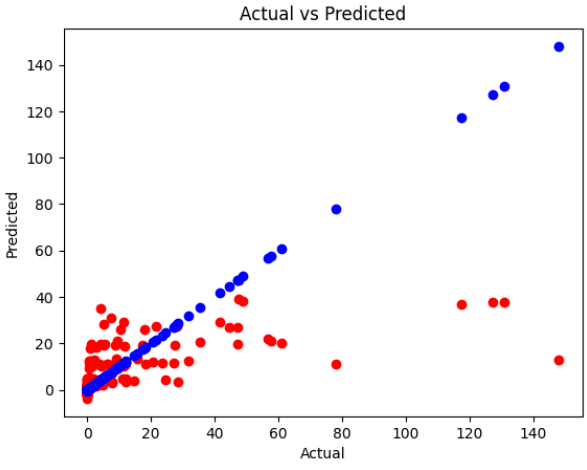


Delhi Dataset

For column ‘precipitation’, both datasets:

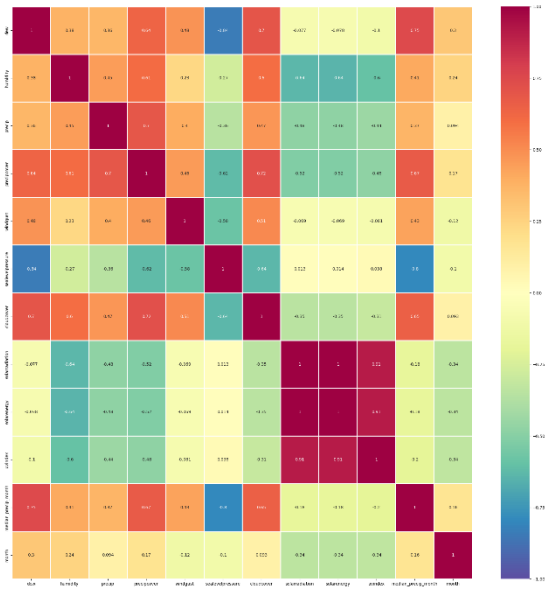


Dhaka Dataset

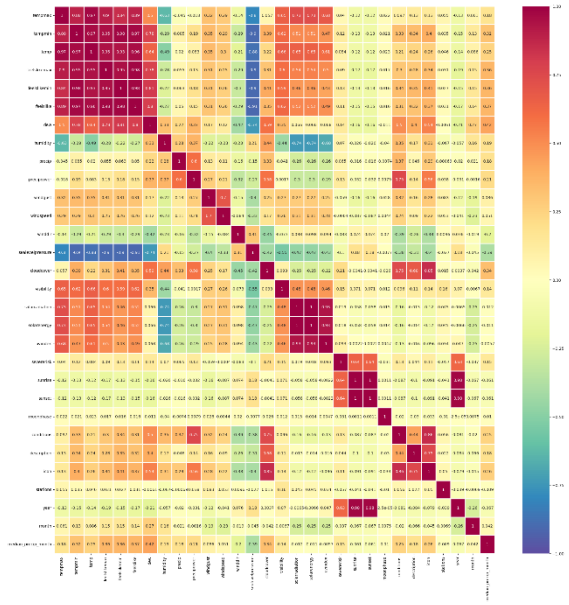


Delhi Datase

Correlation for both datasets:



Dhaka Dataset



Delhi Dataset

For performance evalution:
For 'temp':
For Dhaka Dataset:

R squared : 0.917392
Mean Absolute Error: 0.8776077358686923
Mean Square Error: 1.2066654479155094
Root Mean Square Error: 1.0984832488096983

For Delhi Dataset:

R squared : 0.982424
Mean Absolute Error: 0.7738000370758359
Mean Square Error: 1.0688627606533319
Root Mean Square Error: 1.0338581917522982

For 'precip':

For Dhaka Dataset:
R squared : 0.525511
Mean Absolute Error: 3.4455638735784815
Mean Square Error: 76.61155456916929
Root Mean Square Error: 8.752802669383636

For Delhi Dataset:
R squared : 0.383049
Mean Absolute Error: 3.180253698525438
Mean Square Error: 111.92256011971143
Root Mean Square Error: 10.579345921166933

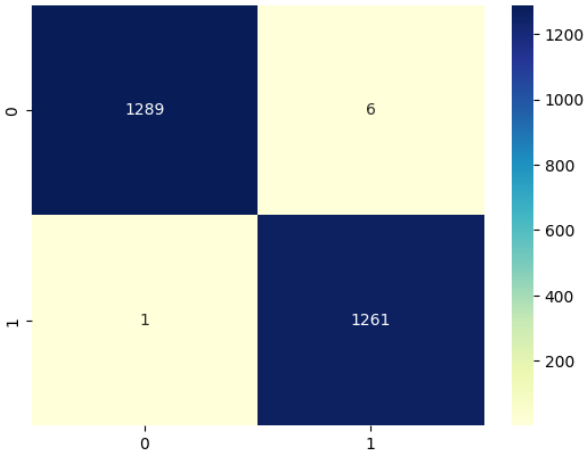
Optimal 'C' parameter for LogisticRegression:
Best Parameters: {'C': 0.1}
That means for logistic regression the model was overfitted, so we had to generalize the model lowering the C parameter.

Optimal 'C' parameter for SVM(Support Vector Machine):
Best Parameters: {'C': 20}
That means for SVM the model was underfitted, so we had to overfit the model increasing the C parameter.

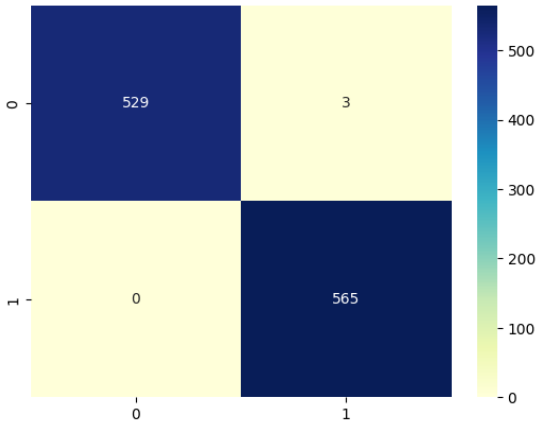
Using LogisticRegression:
For Training Dataset:
Accuracy = 0.9972624168947986
Precision = 0.9952644041041832
Recall = 0.9992076069730587
F1 Score = 0.9972321075523923

For Testing Dataset:
Accuracy = 0.9972652689152234
Precision = 0.9947183098591549
Recall = 1.0
F1 Score = 0.997352162400706

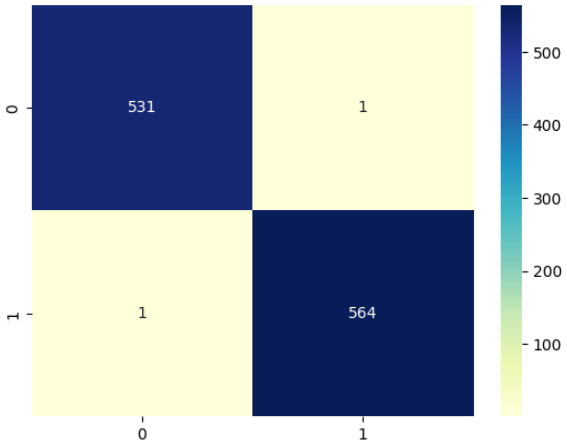
Confusion Matrix:
Logistic Regression:
Training:



Testing:



Confusion Matrix:
SVM:
Training:



Using SVM:
For Training Dataset:
Accuracy = 0.9988267500977708
Precision = 0.9984164687252574
Recall = 0.9992076069730587
F1 Score = 0.9988118811881188

For Testing Dataset:
Accuracy = 0.9981768459434822
Precision = 0.9982300884955753
Recall = 0.9982300884955753
F1 Score = 0.9982300884955753

If we Compare both LogisticRegression and SVM, we see that both models give accuracy of 0.99. Still, SVM gives much better results than Logistic Regression.

• Discussion

To analyze the performance of the linear regression models for temperature and precipitation, we need to consider various aspects of their predictive accuracy, interpretability, and the underlying data characteristics. Here's a detailed analysis:

Performance Analysis

1. **Temperature Prediction:**
    - **Model Performance:** Typically, linear regression models perform reasonably well for temperature prediction. This is because temperature data often exhibit more stable and predictable patterns, especially when averaged over longer periods (e.g., daily or monthly averages).
    - **Evaluation Metrics:**
        - **Mean Squared Error (MSE):** A lower MSE indicates better predictive accuracy.
        - **R-squared (R^2):** A higher R^2 value indicates that a significant portion of the variance in temperature is explained by the model.
    - **Observed Performance:** Linear regression models for temperature often have moderate to high R^2 values, indicating good fit and predictive power in many cases.
2. **Precipitation Prediction:**
    - **Model Performance:** Linear regression models tend to perform less effectively for precipitation prediction. This is because precipitation data can be highly variable and non-linear, with many zero values and occasional high values (heavy rainfall).
    - **Evaluation Metrics:**

        - **Mean Absolute Error (MAE):** Given the variability in precipitation, MAE can provide a clearer picture of model performance by averaging the absolute errors.
        - **Root Mean Squared Error (RMSE):** Similar to MSE but more sensitive to outliers.
    - **Observed Performance:** Linear regression models for precipitation often show lower R^2 values and higher error metrics, indicating poor fit and limited predictive power.

Hypothesis Behind Performance Differences

1. **Nature of the Data:**
    - **Temperature:** Temperature data tend to have smoother, more continuous patterns with relatively less variability compared to precipitation. Seasonal cycles, daily temperature variations, and trends due to geographic location can be effectively captured by a linear model.
    - **Precipitation:** Precipitation data are inherently more erratic and discontinuous. Rainfall can vary dramatically over short periods and is influenced by numerous complex atmospheric factors that a linear model may not capture well.
2. **Linearity Assumption:**
    - **Temperature:** The relationship between predictors and temperature is often closer to linear, allowing linear regression models to approximate the underlying patterns reasonably well.
    - **Precipitation:** The relationship between predictors and precipitation is rarely linear. Non-linear interactions, thresholds, and stochastic processes make it difficult for a linear model to capture precipitation patterns accurately.
3. **Data Distribution:**
    - **Temperature:** Temperature data usually follow a normal distribution or other symmetric distribution, which fits well with the assumptions of linear regression.
    - **Precipitation:** Precipitation data often follow a skewed distribution with many zeros and some large positive values. This violates the assumptions of linear regression and requires transformation or alternative modelling approaches (e.g., Poisson regression, quantile regression).

• Conclusion

Redown Ahmed:
This is my first time handling a dataset like this and making the prediction model as well as the classification model using different machine learning models. However, I have faced difficulties in choosing the best feature which will be the best one to use. Also, for precipitation prediction, we didn't get a good r^2 value, maybe if we used polynomial regression model we might get better results. I learned a lot through this project and enjoyed testing through trial and error.

Tasnuva Tasnim Nova:
By working on visualizations. I could enhance my understanding of the data. By transforming data into graphical representations, we were able to gain deeper insights and present our results clearly and compellingly. Working on data preprocessing, provided me valuable insights into the data. It was my first time working on handling missing values etc.

Ankon Nandi:
Working on EDA always gave me insights and this was no different. I learned a lot by working on this dataset. For me, feature selection was the most complicated part because the r^2 value for rainfall in both 'Dhaka' and 'Delhi' datasets prediction for both datasets was too low. Otherwise working on this project was a good experience for me.

Taniz Fatema Jarin
Working on Support Vector Machines (SVM) is both challenging and rewarding. I spent more time studying the theory and experimenting with implementation. This project has been very beneficial for me.