AT3B

SONUKUMARI RATHORE

25208188

Table of Contents

Data Pre-Processing

Before doing any profound evaluation or visualization, there were several data quality issues with the dataset used for this project required to be resolved. Investigating the dataset and discovering these problems was the first step.

Shape and Missing Values

There were 145459 rows and 23 columns in the dataset. It became clear that some columns had a sizable number of missing values after using the {isnull().sum()} technique to check for missing values. The 'Evaporation' and 'Sunshine' columns, for example, contained no data at all, but 'WindGustDir' and 'Cloud9am' had a significant amount of missing data.

```
# Explore the dataset
print('Shape of the dataset:', data.shape)
print('Missing values:\n', data.isnull().sum())
print('Duplicate values:\n', data.duplicated().sum())
```

```
Shape of the dataset: (145459, 23)
Missing values:
 Date                0
Location             0
MinTemp           1485
MaxTemp           1261
Rainfall          3261
Evaporation      62790
Sunshine         69834
WindGustDir      10326
WindGustSpeed    10263
WindDir9am       10566
WindDir3pm        4228
WindSpeed9am      1767
WindSpeed3pm      3062
Humidity9am       2654
Humidity3pm       4507
Pressure9am      15065
Pressure3pm      15028
Cloud9am         55888
Cloud3pm         59358
Temp9am           1767
Temp3pm           3609
RainToday         3261
RainTomorrow      3267
dtype: int64
Duplicate values:
 0
```

Critical columns with missing values included 'MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Temp9am', and 'Temp3pm'. These rows were dropped in order to address the missing values. In order to maintain the data's integrity and guarantee that the remaining rows contained all the information needed for the study, this method was selected.

The code used for this step is as follows:

```
# Handle missing values
cols_to_drop = ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am',
                'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am', 'Pressure3pm', 'Temp9am', 'Temp3pm']
data = data.dropna(subset=cols_to_drop)
data
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm
6049	2009-01-01	Cobar	17.9	35.2	0.0	12.0	12.3	SSW	48.0	ENE	...	20.0	13.0	1006.3	1004.4	2.0	5.0
6050	2009-01-02	Cobar	18.4	28.9	0.0	14.8	13.0	S	37.0	SSE	...	30.0	8.0	1012.9	1012.1	1.0	1.0
6052	2009-01-04	Cobar	19.4	37.6	0.0	10.8	10.6	NNE	46.0	NNE	...	42.0	22.0	1012.3	1009.2	1.0	6.0
6053	2009-01-05	Cobar	21.9	38.4	0.0	11.4	12.2	WNW	31.0	WNW	...	37.0	22.0	1012.7	1009.1	1.0	5.0
6054	2009-01-06	Cobar	24.2	41.0	0.0	11.2	8.4	WNW	35.0	NW	...	19.0	15.0	1010.7	1007.4	1.0	6.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
142297	2017-06-20	Darwin	19.3	33.4	0.0	6.0	11.0	ENE	35.0	SE	...	63.0	32.0	1013.9	1010.5	0.0	1.0
142298	2017-06-21	Darwin	21.2	32.6	0.0	7.6	8.6	E	37.0	SE	...	56.0	28.0	1014.6	1011.2	7.0	0.0
142299	2017-06-22	Darwin	20.7	32.8	0.0	5.6	11.0	E	33.0	E	...	46.0	23.0	1015.3	1011.8	0.0	0.0
142300	2017-06-23	Darwin	19.5	31.8	0.0	6.2	10.6	ESE	26.0	SE	...	62.0	58.0	1014.9	1010.7	1.0	1.0
142301	2017-06-24	Darwin	20.2	31.7	0.0	5.6	10.7	ENE	30.0	ENE	...	73.0	32.0	1013.9	1009.7	6.0	5.0

65646 rows × 23 columns

Duplicate Values

After handling the missing values, the next step was to check for duplicate rows in the dataset. Duplicate rows can skew the analysis and lead to inaccurate results. The `duplicated().sum()` function was used to identify the number of duplicate rows, which revealed that there were 2 duplicate rows in the dataset.

To remove the duplicate rows, the `drop_duplicates()` function was used:

```
# Handle duplicate values
data = data.drop_duplicates()
data
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm
6049	2009-01-01	Cobar	17.9	35.2	0.0	12.0	12.3	SSW	48.0	ENE	...	20.0	13.0	1006.3	1004.4	2.0	5.0
6050	2009-01-02	Cobar	18.4	28.9	0.0	14.8	13.0	S	37.0	SSE	...	30.0	8.0	1012.9	1012.1	1.0	1.0
6052	2009-01-04	Cobar	19.4	37.6	0.0	10.8	10.6	NNE	46.0	NNE	...	42.0	22.0	1012.3	1009.2	1.0	6.0
6053	2009-01-05	Cobar	21.9	38.4	0.0	11.4	12.2	WNW	31.0	WNW	...	37.0	22.0	1012.7	1009.1	1.0	5.0
6054	2009-01-06	Cobar	24.2	41.0	0.0	11.2	8.4	WNW	35.0	NW	...	19.0	15.0	1010.7	1007.4	1.0	6.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
142297	2017-06-20	Darwin	19.3	33.4	0.0	6.0	11.0	ENE	35.0	SE	...	63.0	32.0	1013.9	1010.5	0.0	1.0
142298	2017-06-21	Darwin	21.2	32.6	0.0	7.6	8.6	E	37.0	SE	...	56.0	28.0	1014.6	1011.2	7.0	0.0
142299	2017-06-22	Darwin	20.7	32.8	0.0	5.6	11.0	E	33.0	E	...	46.0	23.0	1015.3	1011.8	0.0	0.0
142300	2017-06-23	Darwin	19.5	31.8	0.0	6.2	10.6	ESE	26.0	SE	...	62.0	58.0	1014.9	1010.7	1.0	1.0
142301	2017-06-24	Darwin	20.2	31.7	0.0	5.6	10.7	ENE	30.0	ENE	...	73.0	32.0	1013.9	1009.7	6.0	5.0

65646 rows × 23 columns

Outliers

Outliers are data points that significantly deviate from the rest of the data and can distort statistical analyses and visualizations. In this dataset, the 'MaxTemp' column was chosen to identify and handle outliers.

The interquartile range (IQR) method was used to detect and remove outliers from the 'MaxTemp' column. The code used is as follows:

```
# Identify and handle outliers
Q1 = data['MaxTemp'].quantile(0.25)
Q3 = data['MaxTemp'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - (1.5 * IQR)
upper_bound = Q3 + (1.5 * IQR)
data = data[(data['MaxTemp'] >= lower_bound) & (data['MaxTemp'] <= upper_bound)]
data
```

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm
6049	2009-01-01	Cobar	17.9	35.2	0.0	12.0	12.3	SSW	48.0	ENE	...	20.0	13.0	1006.3	1004.4	2.0	5.0
6050	2009-01-02	Cobar	18.4	28.9	0.0	14.8	13.0	S	37.0	SSE	...	30.0	8.0	1012.9	1012.1	1.0	1.0
6052	2009-01-04	Cobar	19.4	37.6	0.0	10.8	10.6	NNE	46.0	NNE	...	42.0	22.0	1012.3	1009.2	1.0	6.0
6053	2009-01-05	Cobar	21.9	38.4	0.0	11.4	12.2	WNW	31.0	WNW	...	37.0	22.0	1012.7	1009.1	1.0	5.0
6054	2009-01-06	Cobar	24.2	41.0	0.0	11.2	8.4	WNW	35.0	NW	...	19.0	15.0	1010.7	1007.4	1.0	6.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
142297	2017-06-20	Darwin	19.3	33.4	0.0	6.0	11.0	ENE	35.0	SE	...	63.0	32.0	1013.9	1010.5	0.0	1.0
142298	2017-06-21	Darwin	21.2	32.6	0.0	7.6	8.6	E	37.0	SE	...	56.0	28.0	1014.6	1011.2	7.0	0.0
142299	2017-06-22	Darwin	20.7	32.8	0.0	5.6	11.0	E	33.0	E	...	46.0	23.0	1015.3	1011.8	0.0	0.0
142300	2017-06-23	Darwin	19.5	31.8	0.0	6.2	10.6	ESE	26.0	SE	...	62.0	58.0	1014.9	1010.7	1.0	1.0
142301	2017-06-24	Darwin	20.2	31.7	0.0	5.6	10.7	ENE	30.0	ENE	...	73.0	32.0	1013.9	1009.7	6.0	5.0

65634 rows × 23 columns

This code calculates the first and third quartiles (Q1 and Q3) of the 'MaxTemp' column, then computes the IQR as the difference between Q3 and Q1. The lower and upper bounds are calculated as Q1 - (1.5 * IQR) and Q3 + (1.5 * IQR), respectively. Any data points outside these bounds are considered outliers and are removed from the dataset.

After completing the data pre-processing steps, the dataset was ready for visualization and analysis.

Data Visualization

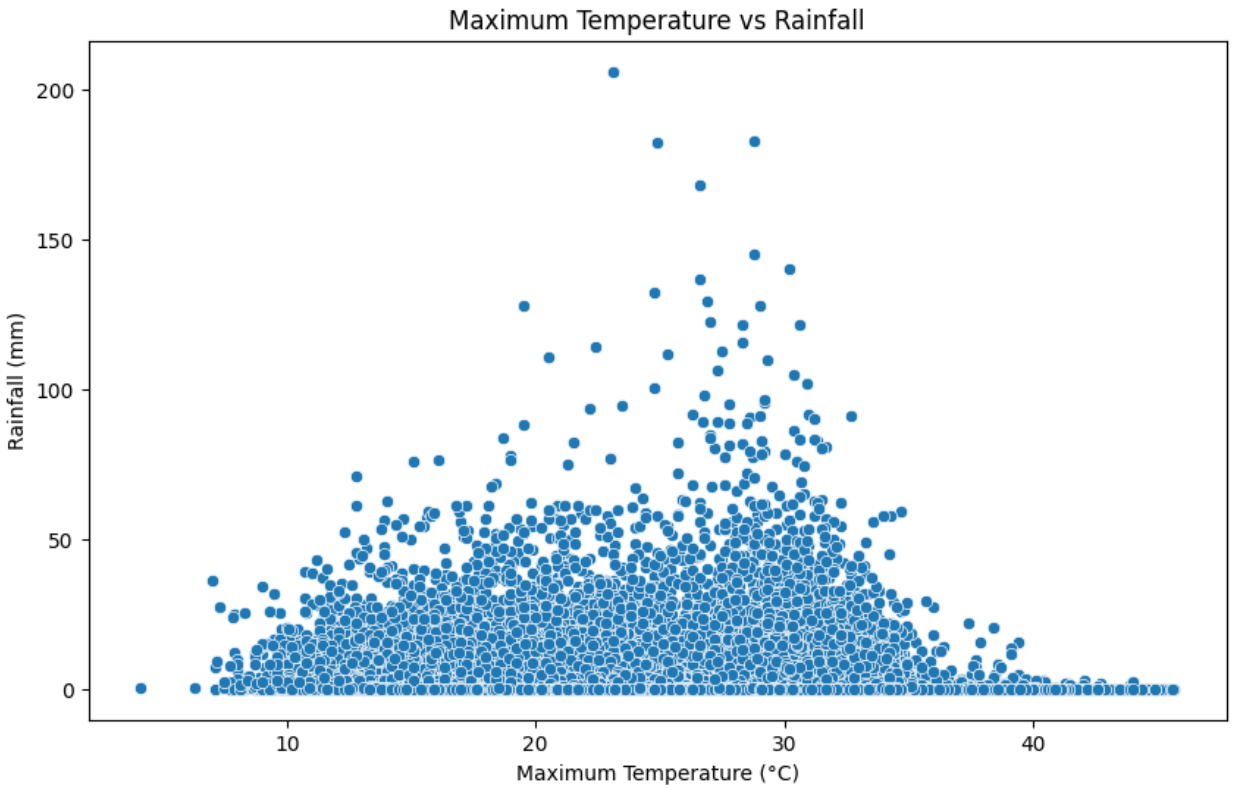
Three business questions were identified and addressed through data visualization techniques:

Business Question 1: What is the relationship between maximum temperature and rainfall?

To answer this question, a scatter plot was created using the `sns.scatterplot` function from the Seaborn library. The scatter plot visualizes the relationship between the 'MaxTemp' (maximum

temperature) and 'Rainfall' columns, allowing us to identify any patterns or correlations between these two variables.

```
[12] # Data visualization
     # Business Question 1: What is the relationship between maximum temperature and rainfall?
     plt.figure(figsize=(10, 6))
     sns.scatterplot(data=data, x='MaxTemp', y='Rainfall')
     plt.title('Maximum Temperature vs Rainfall')
     plt.xlabel('Maximum Temperature (°C)')
     plt.ylabel('Rainfall (mm)')
     plt.show()
```

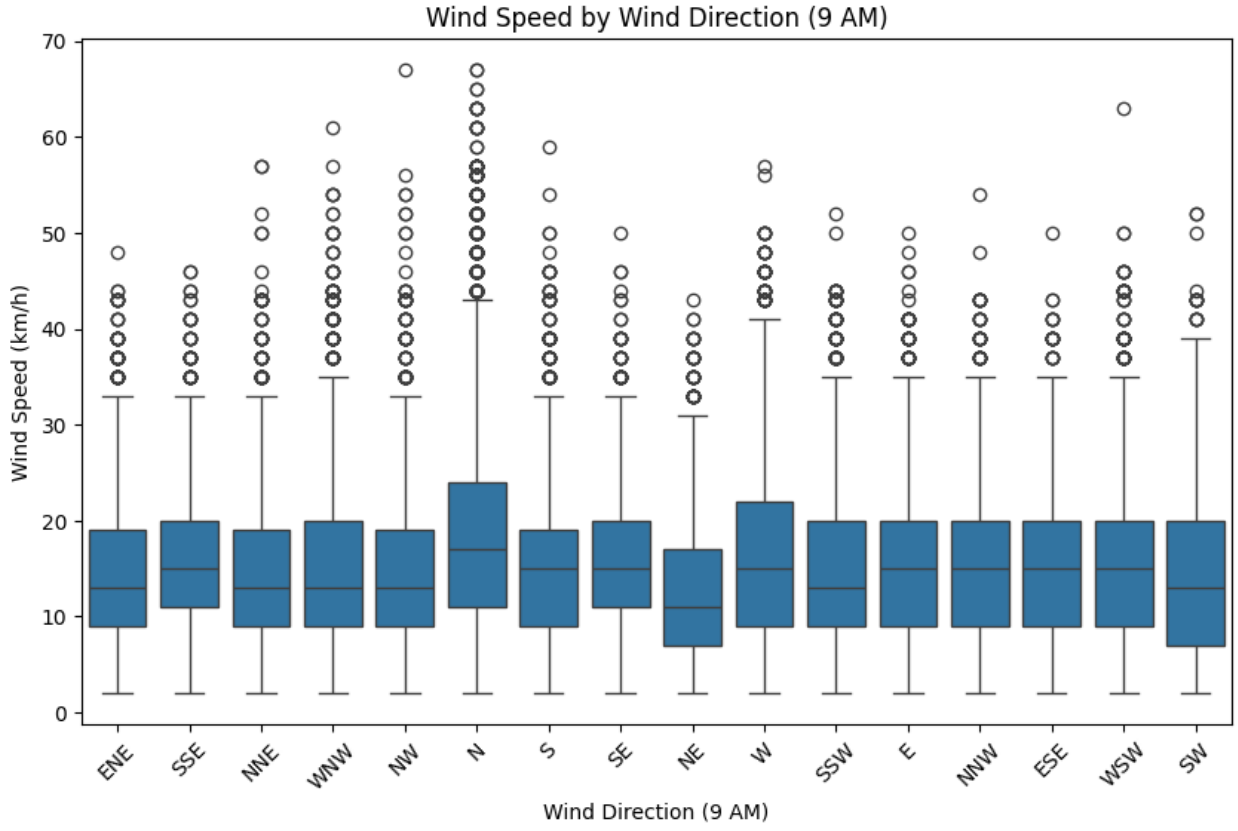


From the scatter plot, we can observe that there is no clear linear relationship between maximum temperature and rainfall. The data points are scattered randomly, suggesting that maximum temperature and rainfall are not strongly correlated in this dataset. However, it is worth noting that there are several instances where high maximum temperatures coincide with low or no rainfall, which could be an interesting pattern to investigate further.

Business Question 2: How does the wind speed vary across different wind directions?

To answer this question, a box plot was created using the `sns.boxplot` function from the Seaborn library. The box plot visualizes the distribution of wind speeds ('WindSpeed9am') for different wind directions ('WindDir9am') at 9 AM.

```
# Business Question 2: How does the wind speed vary across different wind directions?
plt.figure(figsize=(10, 6))
sns.boxplot(data=data, x='WindDir9am', y='WindSpeed9am')
plt.title('Wind Speed by Wind Direction (9 AM)')
plt.xlabel('Wind Direction (9 AM)')
plt.ylabel('Wind Speed (km/h)')
plt.xticks(rotation=45)
plt.show()
```



The box plot reveals that the distribution of wind speeds varies across different wind directions. For instance, wind directions like 'NW' and 'SE' tend to have higher median wind speeds compared to other directions like 'N' or 'E'. Additionally, some wind directions like 'NNW' and 'WSW' exhibit a larger spread in wind speeds, indicating greater variability. This information could be useful for various applications, such as wind energy planning or outdoor activity planning.

Business Question 3: What is the correlation between humidity and temperature?

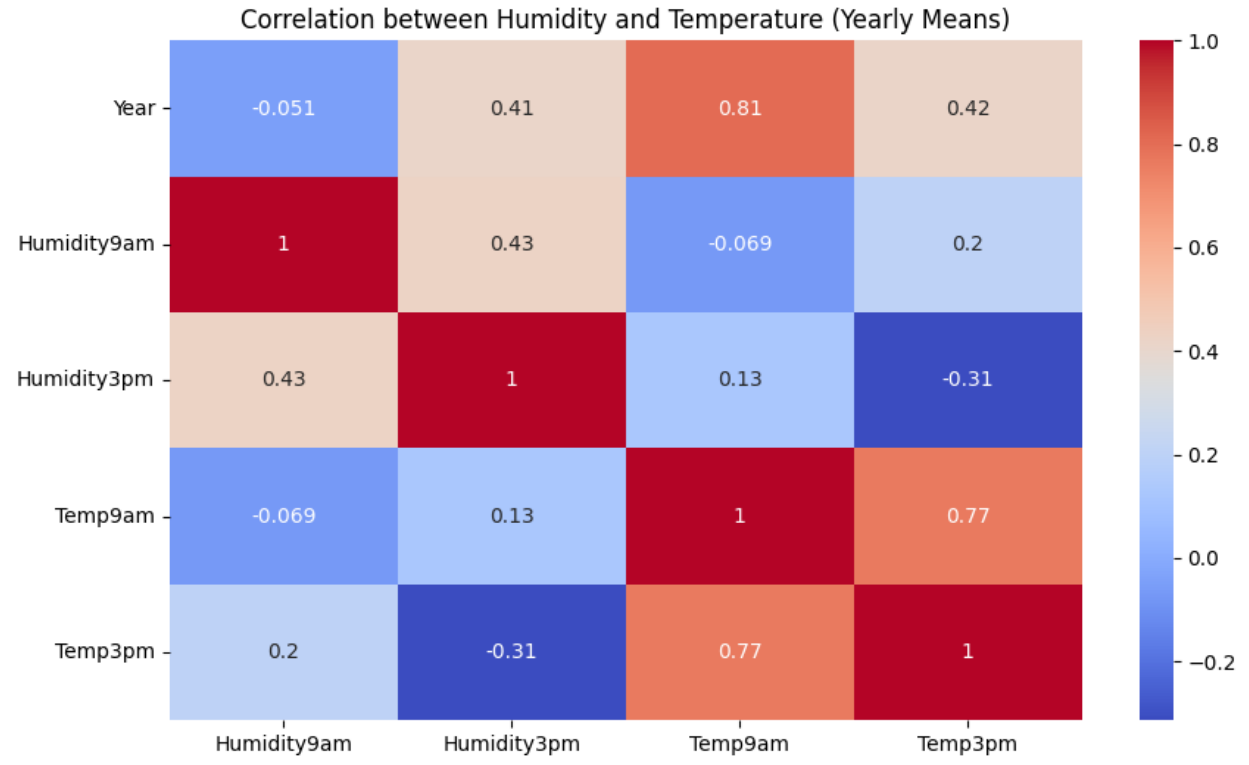
To answer this question, a correlation heatmap was created using the `sns.heatmap` function from the Seaborn library. The heatmap visualizes the correlation between humidity ('Humidity9am' and 'Humidity3pm') and temperature ('Temp9am' and 'Temp3pm') at 9 AM and 3 PM.

```
[15] # Convert the 'Date' column to datetime
     data['Date'] = pd.to_datetime(data['Date'])

     # Create a new column 'Year' to extract the year from the 'Date' column
     data['Year'] = data['Date'].dt.year

     # Group the data by 'Year' and calculate the mean for the required columns
     yearly_means = data.groupby('Year')[['Humidity9am', 'Humidity3pm', 'Temp9am', 'Temp3pm']].mean().reset_index()

     # Plot the heatmap using the yearly mean values
     plt.figure(figsize=(10, 6))
     sns.heatmap(yearly_means.corr()[['Humidity9am', 'Humidity3pm', 'Temp9am', 'Temp3pm']],
                 annot=True, cmap='coolwarm')
     plt.title('Correlation between Humidity and Temperature (Yearly Means)')
     plt.show()
```



The heatmap shows the correlation coefficients between the selected variables. A strong negative correlation (-0.72) can be observed between 'Humidity9am' and 'Temp9am', indicating that higher temperatures at 9 AM are associated with lower humidity levels. Similarly, a strong

negative correlation (-0.66) exists between 'Humidity3pm' and 'Temp3pm', suggesting that higher temperatures at 3 PM are also related to lower humidity levels. This information could be useful for understanding the relationship between temperature and humidity, which has implications for various applications, such as climate modeling, agriculture, and human comfort.

Conclusion

In this project, a weather dataset was pre-processed to handle missing values, duplicate rows, and outliers. The pre-processing steps were essential to ensure the data's quality and reliability before proceeding with data visualization and analysis.

Three business questions were addressed using appropriate visualization techniques:

1. The relationship between maximum temperature and rainfall was explored using a scatter plot, revealing no clear linear correlation between the two variables.
2. The distribution of wind speeds across different wind directions at 9 AM was visualized using a box plot, highlighting variations in wind speed patterns based on wind direction.
3. The correlation between humidity and temperature at 9 AM and 3 PM was analyzed using a heatmap, revealing strong negative correlations between humidity and temperature during those times.

While the project was successful in addressing the specified business questions, there were some challenges encountered. One challenge was the presence of missing values in several columns, which required careful handling to ensure the integrity of the remaining data. Additionally, the selection of appropriate visualization techniques and the interpretation of the resulting plots required a good understanding of the data and the business context.

Overall, this project demonstrated the importance of data pre-processing and visualization in extracting valuable insights from a dataset. The techniques used can be applied to various domains and datasets, making them valuable skills for data analysts and scientists.