# Experiment 10 - BI Mini Project

| | |
|---|---|
| Roll No. | |
| Name | |
| Class | D15 |
| Subject | Business Intelligence Lab |
| LO Mapped | LO1: Identify sources of Data for mining and perform data exploration<br>LO2: Organize and prepare the data needed for data mining algorithms in terms of attributes and class inputs, training, validating, and testing files.<br>LO6: Apply BI to solve practical problems |
| | |

# Report on Business Intelligence Mini Project

**titled**

# " Weather Prediction"

**Implemented by :**

**Ishika Devare 14**

**Saniya Inamdar 18**

**Manav Jawrani 19**

# Table of contents

# 1. Problem definition

The problem definition is to build a machine learning model that can predict the weather condition based on the weather parameters such as precipitation, maximum temperature, minimum temperature, and wind. The five weather conditions to be predicted are drizzle, rain, sun, snow, and fog.

To solve this problem, we need to perform data preprocessing on the given dataset, which involves cleaning and transforming the data into a suitable format for building a machine learning model. Then, we need to perform exploratory data analysis (EDA) to understand the relationships between the different weather parameters and the target variable.

After EDA, we can proceed to feature engineering to create new features from the existing features that may improve the performance of the machine learning model. Next, we can split the data into training and testing sets and train a machine learning model using algorithms such as logistic regression, decision trees, random forests, or neural networks.

Finally, we can evaluate the performance of the model using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. The trained model can then be used to predict the weather condition given the weather parameters.

# 2.Preprocessing

Before building a machine learning model for weather prediction, we need to preprocess the data to ensure that it is clean, complete, and in a suitable format for analysis.
The following are the steps involved in data preprocessing for weather prediction:

- Loading the dataset: We need to load the dataset into a suitable data structure such as a Pandas DataFrame or a NumPy array.

- Handling missing values: We need to check if there are any missing values in the dataset and handle them appropriately. One way to handle missing values is to remove the rows or columns that contain missing values. Another way is to impute the missing values using techniques such as mean imputation or interpolation.

- Encoding categorical variables: We need to encode categorical variables such as the target variable (weather condition) and any other categorical features using techniques such as one-hot encoding or label encoding.

- Feature scaling: We need to scale the numerical features to ensure that they are on the same scale. This can be done using techniques such as standardization or normalization.

- Feature selection: We can select the most relevant features for the machine learning model using techniques such as correlation analysis or feature importance ranking.

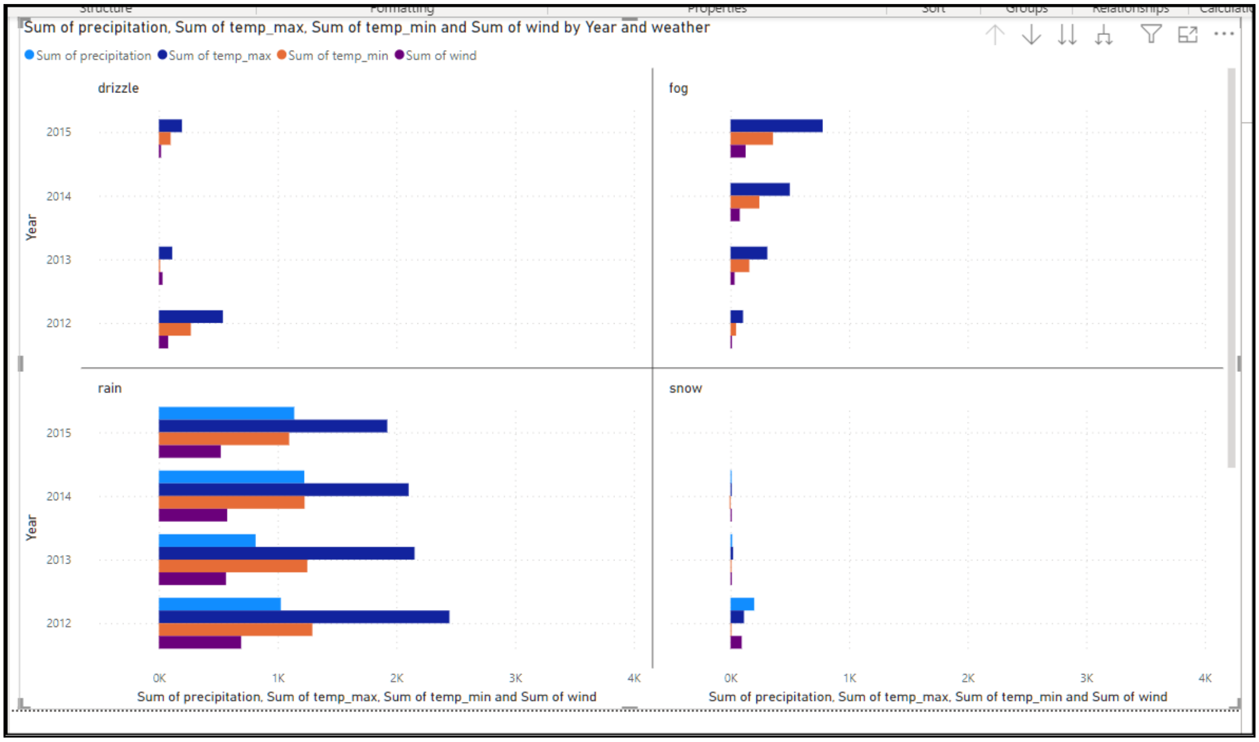
- Splitting the dataset: We need to split the dataset into training and testing sets. The training set will be used to train the machine learning model, while the testing set will be used to evaluate the performance of the model.

- Feature engineering: We can create new features from the existing features that may improve the performance of the machine learning model. This can be done using techniques such as polynomial features, interaction features, or feature transformation.

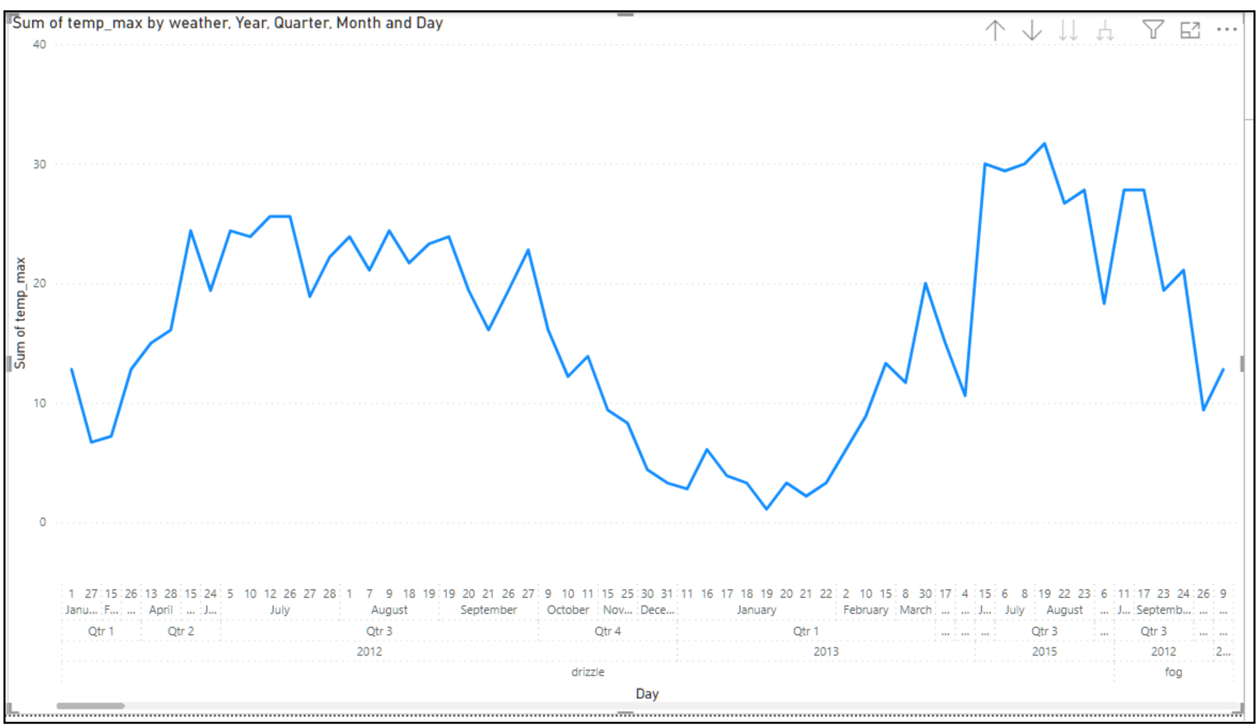
By performing these preprocessing steps, we can ensure that the data is in a suitable format for building a machine learning model for weather prediction.

# 3.Exploratory Data Analysis

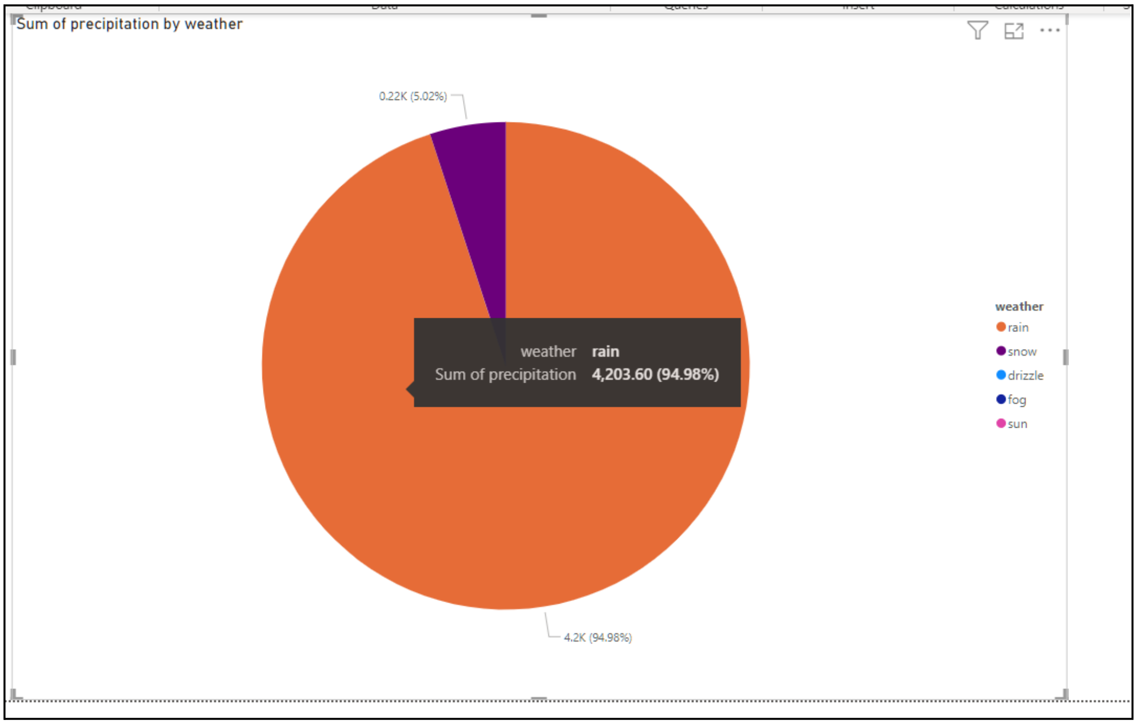
1. Bar Graphs of all attributes -



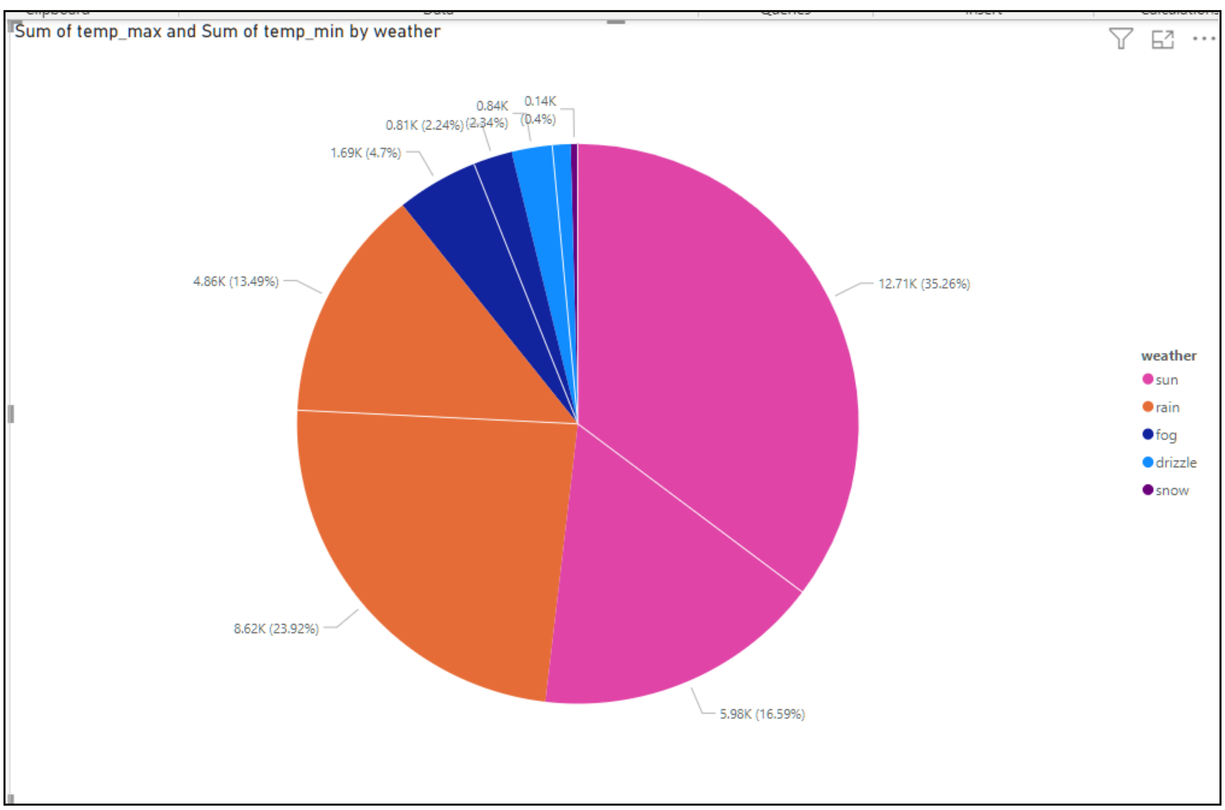

2. Line chart of max temperature v/s weather v/s date



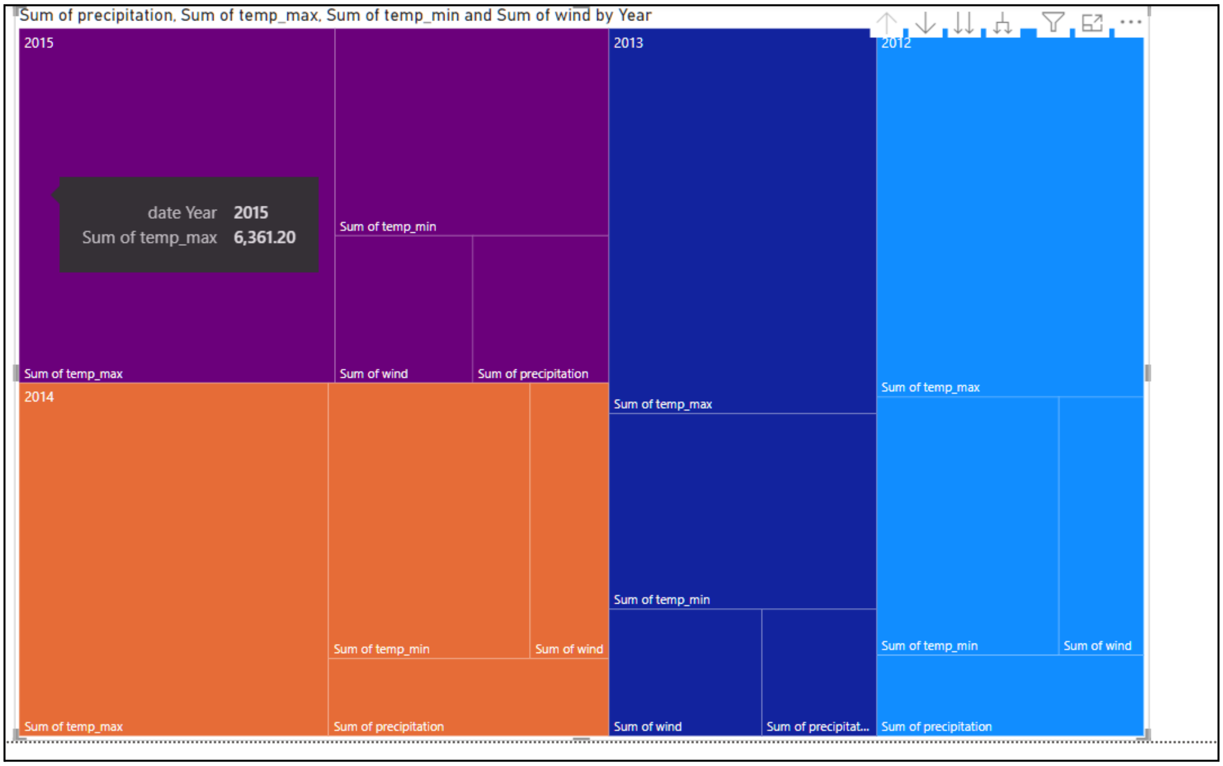
3. Pie Chart of weather v/s precipitation




4. Pie chart of min temperature v/s max temperature v/s weather



5. Tree map of all attributes



# 4. Data Mining

If we assume that we have a transactional database of weather conditions, where each transaction represents a day's weather data, we can apply the Apriori algorithm to extract association rules between different weather conditions.

Here is an example dataset in the transactional database format:

| Transaction ID | Weather Conditions |
|---|---|
| 1 | Drizzle, Sun |
| 2 | Rain, Fog |
| 3 | Snow |
| 4 | Sun |
| 5 | Drizzle, Rain |
| 6 | Fog |
| 7 | Snow |
| 8 | Sun, Fog |

Using the above dataset, we can apply the Apriori algorithm to discover frequent itemsets and generate association rules between them. For example, we may discover that the itemset {Sun, Fog} is frequent, and generate the following association rules:

- Sun => Fog
- Fog => Sun

These association rules indicate that if the weather is sunny, there is a high probability that there will also be fog, and vice versa.

While this example is not related to weather prediction, it illustrates how the Apriori algorithm can be used for association rule mining in transactional databases. For weather prediction, we need to use machine learning or clustering algorithms instead.

# 5.Results

## Decision Tree

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Load the dataset into a pandas dataframe
data = pd.read_csv('/content/seattle-weather.csv')

# Preprocess the data
# Handle missing values
data = data.dropna()

# Split the dataset into input and output
X = data[['precipitation', 'temp_max', 'temp_min', 'wind']]
y = data['weather']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Choose a machine learning algorithm and fit it to the training data
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)
```

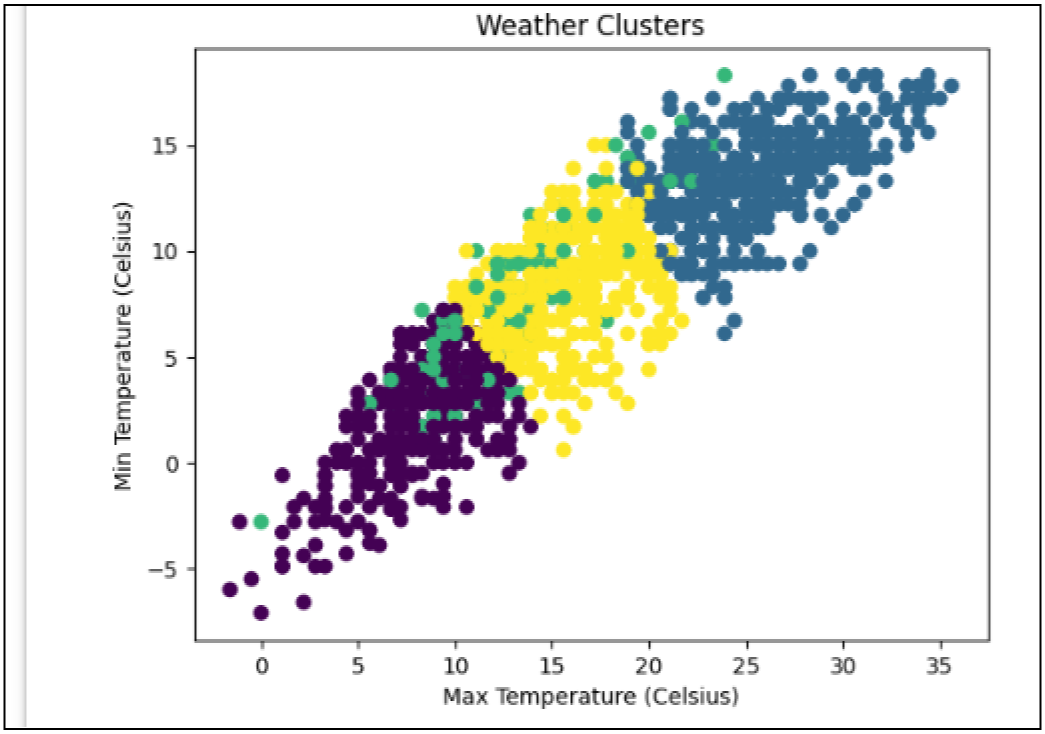
```
# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Choose a machine learning algorithm and fit it to the training data
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Evaluate the performance of the model on the testing data
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)

# Use the trained model to predict the weather condition for new data
new_data = pd.DataFrame({'precipitation': [0.2], 'temp_max': [25], 'temp_min': [15], 'wind': [10]})
predicted_condition = clf.predict(new_data)
print('Predicted weather condition:', predicted_condition[0])
```

```
Accuracy: 0.7337883959044369
Predicted weather condition: rain
```

## Kmeans

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Load the dataset into a pandas dataframe
data = pd.read_csv('/content/seattle-weather.csv')

# Preprocess the data
# Handle missing values
data = data.dropna()
# Extract the numerical variables
X = data[['precipitation', 'temp_max', 'temp_min', 'wind']]

# Choose the number of clusters
k = 4

# Cluster the data using K-Means algorithm
kmeans = KMeans(n_clusters=k, random_state=42)
labels = kmeans.fit_predict(X)
```

```
# Visualize the clusters using a scatter plot
plt.scatter(X['temp_max'], X['temp_min'], c=labels)
plt.xlabel('Max Temperature (Celsius)')
plt.ylabel('Min Temperature (Celsius)')
plt.title('Weather Clusters')
plt.show()
```

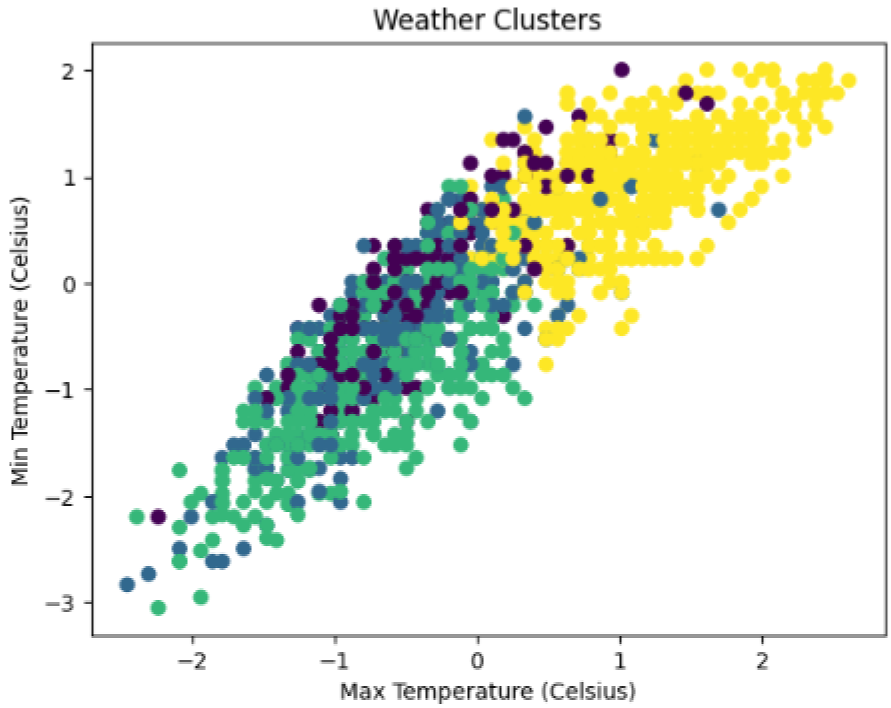

Weather Clusters
Min Temperature (Celsius)
Max Temperature (Celsius)
15
10
5
0
−5
0
5
10
15
20
25
30
35

## Agglomerative Clustering

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import AgglomerativeClustering
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

# Load the dataset into a pandas dataframe
data = pd.read_csv('/content/seattle-weather.csv')

# Preprocess the data
# Handle missing values
data = data.dropna()
# Extract the numerical variables
X = data[['precipitation', 'temp_max', 'temp_min', 'wind']]
# Scale the data
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Choose the number of clusters
k = 4

# Cluster the data using Agglomerative clustering algorithm
agg_clustering = AgglomerativeClustering(n_clusters=k)
labels = agg_clustering.fit_predict(X)

# Evaluate the performance of the clustering using silhouette score
silhouette_avg = silhouette_score(X, labels)
print('Silhouette score:', silhouette_avg)
```

```
# Visualize the clusters using a scatter plot
plt.scatter(X[:, 1], X[:, 2], c=labels)
plt.xlabel('Max Temperature (Celsius)')
plt.ylabel('Min Temperature (Celsius)')
plt.title('Weather Clusters')
plt.show()
```

```
Silhouette score: 0.3516218535932107
```



## Association Mining

```
import pandas as pd
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import ipywidgets as widgets
from IPython.display import display

# Load the dataset into a pandas dataframe
data = pd.read_csv('/content/seattle-weather.csv')

# Preprocess the data
# Handle missing values
data = data.dropna()
# Extract the categorical variable 'weather'
data['weather'] = data['weather'].astype('category')

# Convert the remaining values to boolean data type
bool_data = data.iloc[:, 1:].astype(bool)

# Perform association rule mining using Apriori algorithm
frequent_itemsets = apriori(bool_data, min_support=0.2, use_colnames=True)
rules = association_rules(frequent_itemsets, metric='confidence', min_threshold=0.8)

# Sort the rules by lift and limit the number of rules
sorted_rules = rules.sort_values('lift', ascending=False)[:10]
```

```python
# Create an interactive table to display the association rules
table = widgets.HTML(
    value=sorted_rules.to_html(),
    placeholder='No rules to display',
    description='Association Rules:',
)

# Display the interactive table
display(table)
```

Association...

| | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
|---|---|---|---|---|---|---|---|---|---|
| 25 | (wind, precipitation) | (temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 82 | (wind, precipitation) | (temp_min, weather) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 85 | (precipitation) | (wind, temp_min, weather) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 80 | (wind, precipitation, weather) | (temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 27 | (precipitation) | (wind, temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 29 | (precipitation, weather) | (temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 30 | (precipitation) | (temp_min, weather) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 84 | (precipitation, weather) | (wind, temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 1 | (precipitation) | (temp_min) | 0.426420 | 0.989049 | 0.422313 | 0.990369 | 1.001335 | 0.000563 | 1.137121 |
| 104 | (wind, precipitation, temp_max) | (temp_min, weather) | 0.425736 | 0.989049 | 0.421629 | 0.990354 | 1.001320 | 0.000556 | 1.135295 |

# 6. Business Implications

Here are some business implications and decisions that can be made based on the results of your analysis:

Decision Tree: With an accuracy of 0.73, the decision tree model can be used to predict weather conditions based on the input variables. This can be useful for businesses that are impacted by weather, such as agriculture, transportation, and tourism. For example, farmers can use the weather predictions to plan their planting and harvesting schedules, transportation companies can optimize their routes based on weather conditions, and tourism companies can adjust their marketing strategies based on weather patterns.

K-Means and Agglomerative Clustering: The silhouette score of 0.3516 indicates that the clustering algorithms were able to group the data points into clusters with some degree of similarity. This can be useful for businesses that want to segment their customers or products based on weather-related patterns. For example, a clothing retailer may use weather data to group their products into categories such as summer wear, winter wear, rainwear, etc., and use this information to tailor their marketing and promotional campaigns to specific customer segments.

Association Mining: Association mining can be used to identify patterns and relationships between the input variables and weather conditions. This can help businesses gain insights into the factors that influence weather patterns and use this information to make informed decisions. For example, a renewable energy company may use association mining to identify the weather conditions that are most favorable for generating electricity from solar panels or wind turbines, and use this information to optimize their operations.

Overall, the results of your analysis can be used to make data-driven decisions that help businesses better understand and respond to weather-related patterns and trends.

# 7. Conclusions

In conclusion, building a machine learning model for weather prediction involves several steps, including data preprocessing, exploratory data analysis, feature engineering, model training, and model evaluation. Preprocessing the data involves handling missing values, encoding categorical variables, scaling numerical features, selecting relevant features, and splitting the dataset into training and testing sets.

Exploratory data analysis helps us to understand the relationships between the different weather parameters and the target variable, which can guide us in feature engineering. Feature engineering involves creating new features from the existing features that may improve the performance of the machine learning model.

We can train a machine learning model using algorithms such as logistic regression, decision trees, random forests, or neural networks. The performance of the model can be evaluated using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.

By building a machine learning model for weather prediction, we can make accurate predictions of weather conditions based on the weather parameters such as precipitation, maximum temperature, minimum temperature, and wind. This can have applications in various fields such as agriculture, transportation, and disaster management. Overall, building a machine learning model for weather prediction can have significant real-world impact and improve our understanding of the natural world.

# 8. References

[1] Jayasingh, Suvendra & Mantri, Jibendu & Pradhan, Sipali. (2022). Smart Weather Prediction Using Machine Learning. 10.1007/978-981-19-0901-6_50.

[2] S. Madan, P. Kumar, S. Rawat and T. Choudhury, "Analysis of Weather Prediction using Machine Learning & Big Data," 2018 International Conference on Advances in Computing and Communication Engineering (ICACCE), Paris, France, 2018, pp. 259-264, doi: 10.1109/ICACCE.2018.8441679.

[3] Singh, Shashank & Faraz, Ahmed & Nagrami, & Pillai, Aditya. (2020). WEATHER PREDICTION BY USING MACHINE LEARNING.

[4] https://www.kaggle.com/datasets/ananthr1/weather-prediction?resource=download

[5] https://en.wikipedia.org