Weather Type Prediction Using ANN Model

Problem Statement:

The aim is to predict the Type of the Weather using weather attributes.

Dataset Overview:

The dataset consists of 13,200 fossil samples with 11 features:

Temperature (numeric): The temperature in degrees Celsius, ranging from extreme cold to extreme heat.

Humidity (numeric): The humidity percentage, including values above 100% to introduce outliers.

Wind Speed (numeric): The wind speed in kilometers per hour, with a range including unrealistically high values.

Precipitation (%) (numeric): The precipitation percentage, including outlier values.

Cloud Cover (categorical): The cloud cover description.

Atmospheric Pressure (numeric): The atmospheric pressure in hPa, covering a wide range.

UV Index (numeric): The UV index, indicating the strength of ultraviolet radiation.

Season (categorical): The season during which the data was recorded.

Visibility (km) (numeric): The visibility in kilometers, including very low or very high values.

Location (categorical): The type of location where the data was recorded.

Data Pre-Processing:

Essential libraries such as NumPy and Pandas for data manipulation, and Matplotlib and Seaborn for visualization, were imported. The dataset was copied to ensure original data integrity, and checks for data size, shape, and correctness of column names and types were conducted. No null values or duplicates were found. The dataset was then split into categorical and numeric variables for detailed univariate analysis.

Univariate Analysis:

Categorical Data Analysis:

The initial phase focused on examining categorical data by iterating through each categorical column to identify subclasses and determine their distribution within the dataset. This process provided insights into the diversity and frequency of subclasses present, essential for understanding their impact on the target variable. Our target variable is Categorical, so when we look at the value counts of target variable we found out that the counts of subclass in the target variable are equal that means there is no imbalanced data for us.

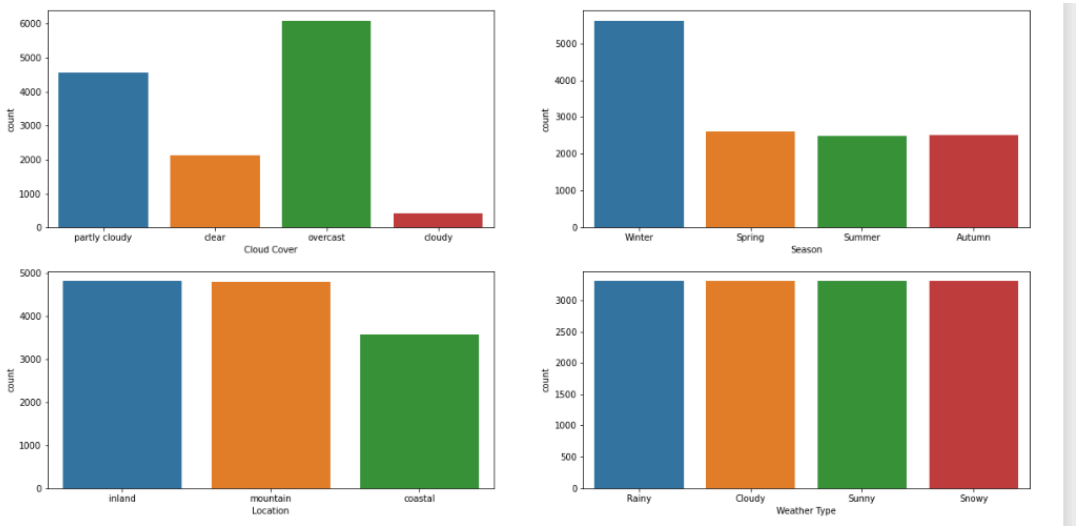
```
Cloud Cover
overcast         6090
partly cloudy    4560
clear            2139
cloudy            411
Name: Cloud Cover, dtype: int64
Season
Winter    5610
Spring    2598
Autumn    2500
Summer    2492
Name: Season, dtype: int64
Location
inland      4816
mountain    4813
coastal     3571
Name: Location, dtype: int64
Weather Type
Rainy     3300
Cloudy    3300
Sunny     3300
Snowy     3300
Name: Weather Type, dtype: int64
```

Numeric Data Analysis:

Attention then shifted to numeric columns. Box plots were generated to assess the distribution and presence of outliers in each numeric attribute. This step revealed outliers in certain columns, highlighting the need for outlier treatment to enhance the reliability of subsequent analyses and model performance. And Distribution plot were used to see the distribution of the data.

Count Plot Analysis:

Count plots were generated for each categorical column to visualize the spread and frequency of subclasses. This facilitated a deeper understanding of how categorical attributes contribute to the dataset's composition and variability. We found out so many insights got from the count plots.



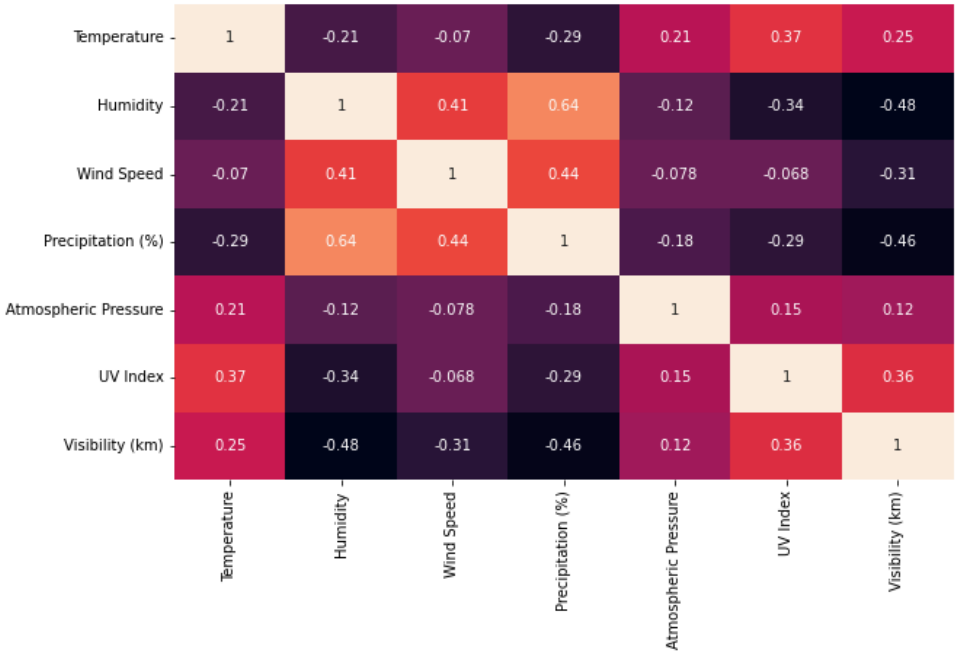
Comprehensive Analysis:

Each column's characteristics, spread, and value distributions were thoroughly evaluated within the entire dataset context. Detailed visual representations are best explored in the accompanying notebook for clarity and precision.

Bi-Variate Analysis:

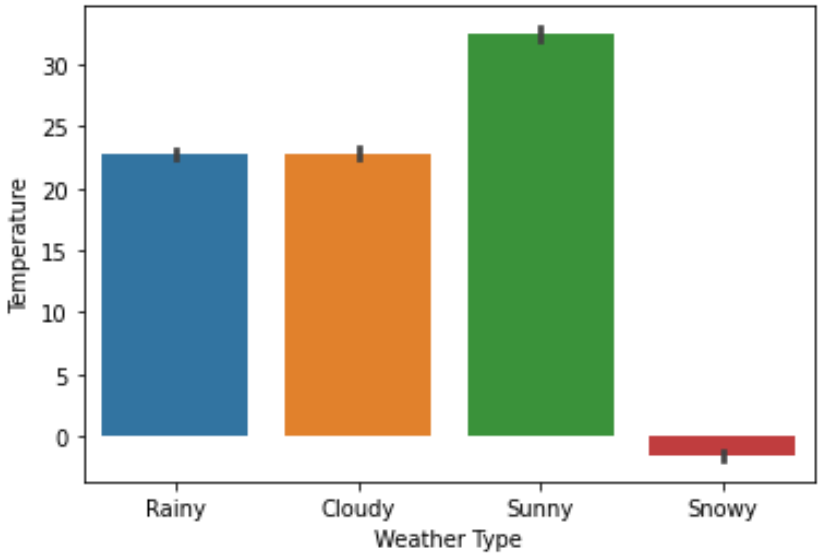
Heatmap of Correlation:

A heatmap was generated to visualize correlations between variables in the dataset, focusing on their impact on the target variable. The analysis revealed no significant multicollinearity issues among the predictors. As our target variable is categoric we cannot plot it in correlation map.



Bar plot for each Numeric Column with Target Column:

Plotting Bar plot for each numeric column present in the data with respect to our Target column to see how each column is contributing in the target sub classes. As Temperature, Humidity .etc are use full to each sub class in the target column.



Frequency Table for each Categoric variable with Target Column:

As Plots are not suitable to tell the relation between variables when both are Categoric Variables. In these cases Crosstab works well, as it a table of frequency of occur.

Cloud Cover Cloud Cover Weather Type	clear	cloudy	overcast	partly cloudy
Cloudy	0	92	1305	1903
Rainy	0	105	2193	1002
Snowy	0	107	2489	704
Sunny	2139	107	103	951

Season Season Weather Type	Autumn	Spring	Summer	Winter
Cloudy	806	850	766	878
Rainy	796	831	820	853
Snowy	69	80	65	3086
Sunny	829	837	841	793

Location Location Weather Type	coastal	inland	mountain
Cloudy	1106	1107	1087
Rainy	1216	1069	1015
Snowy	120	1575	1605
Sunny	1129	1065	1106

Outlier Analysis and Treatment:

Outlier analysis involved checking if outliers in a particular column have any relation with other columns. Pair plots and value counts for numerical columns and categorical columns were used for this purpose respectively. It was found that outliers in numeric columns are related to other variables; hence, outlier treatment was not performed.

Feature Scaling:

Given that the numeric variables do not follow a normal distribution, Min-Max Scaling was applied instead of Standard Scaling. This approach was chosen based on the range of values for each variable.

Model Building:

Two Artificial Neural Network (ANN) models were built: one without scaled values and one with scaled values. The first ANN model without Scaled values are taken to built this model. The model starts with a Flatten layer to reshape the input data, followed by two Dense layers with 256 and 128 neurons respectively, each with a ‘ReLU’ activation function and ‘BatchNormalization’ to improve training stability and speed. The final layer is a Dense layer with 4 neurons and a ‘softmax’ activation function, which outputs a probability distribution over the four classes. The model is compiled with the Adam optimizer and the categorical cross-entropy loss function, which is appropriate for multi-class classification problems. The model is then trained using the fit() method on the training data, with a batch size of 50 and for 50 epochs. The validation data is used to evaluate the model's performance during

training and help prevent overfitting. Overall, this code implements a standard feedforward neural network architecture for multi-class classification, using best practices such as BatchNormalization and appropriate loss functions. By this model we got the accuracy score of 80%. Let's See how the model work with Scaled Values.

```
loss, accuracy = model.evaluate(xtest, ytest)
print(f'Test Accuracy: {accuracy:.2f}')
```

```
83/83 ━━━━━━━━━━━━━━━━━━━━ 0s 580us/step - accuracy: 0.8017 - loss: 0.5824
Test Accuracy: 0.80
```

First, the input data is preprocessed by dropping the 'Weather Type' column from the scaled dataframe and separating the remaining columns into input features (x) and target labels (y). The data is then split into training and testing sets using the train_test_split function from Scikit-learn. The target labels are one-hot encoded using the to_categorical function from Keras, which converts the labels into a binary matrix representation with 4 columns (one for each weather type).

Next, a Sequential model is created and the first layer is added using the Flatten function to flatten the input data into a 1D array. A Dense layer with 256 neurons and ReLU activation function is added next, followed by BatchNormalization to improve training stability and speed. Another Dense layer with 128 neurons and ReLU activation is added, followed by another BatchNormalization layer. The final layer is a Dense layer with 4 neurons and a softmax activation function, which outputs a probability distribution over the four weather types.

The model is then compiled using the Adam optimizer, categorical cross-entropy loss function, and accuracy metric. Finally, the model is trained using the fit function on the training data for 50 epochs with a batch size of 50, and the validation data is used to evaluate the model's performance during training. The output of the model is a trained neural network that can be used to predict the weather type based on the input features. This Model performed well for our data with an accuracy of 90% and here you can see the classification report.

	precision	recall	f1-score	support
0	0.86	0.90	0.88	632
1	0.93	0.86	0.90	682
2	0.93	0.93	0.93	699
3	0.97	0.88	0.92	627
micro avg	0.92	0.89	0.91	2640
macro avg	0.92	0.89	0.91	2640
weighted avg	0.92	0.89	0.91	2640
samples avg	0.89	0.89	0.89	2640