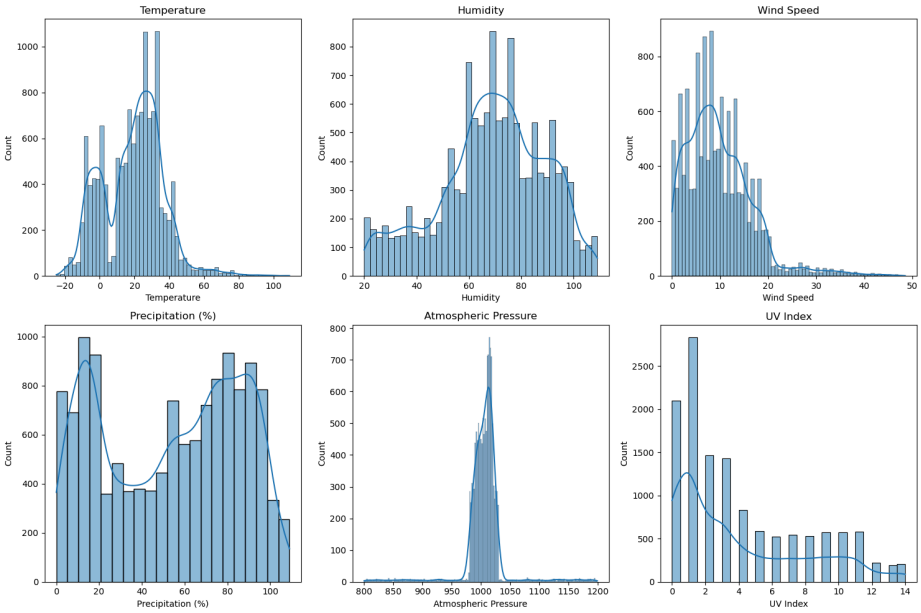
Background

This analysis is done on a dataset of simulated weather conditions, including various features such as temperature, humidity, wind speed, precipitation, atmospheric pressure, UV index, visibility, cloud cover, season, location, and weather type labels. The goal is to develop a predictive model to classify weather types based on these features. These results can be used to show the effectiveness of different machine learning algorithms and how they can be implemented to provide more accurate weather forecasts.

Exploratory Data Analysis

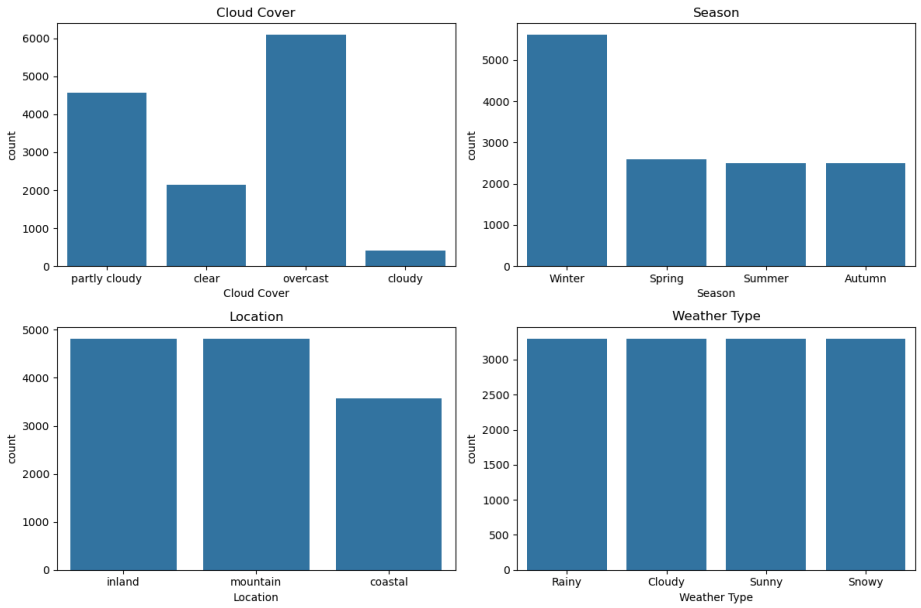
Numerical Attribute Distributions:



Observations:

The numerical attributes have a range of distributions, with Atmospheric Pressure being the only attribute to have a seemingly 'normal' distribution.

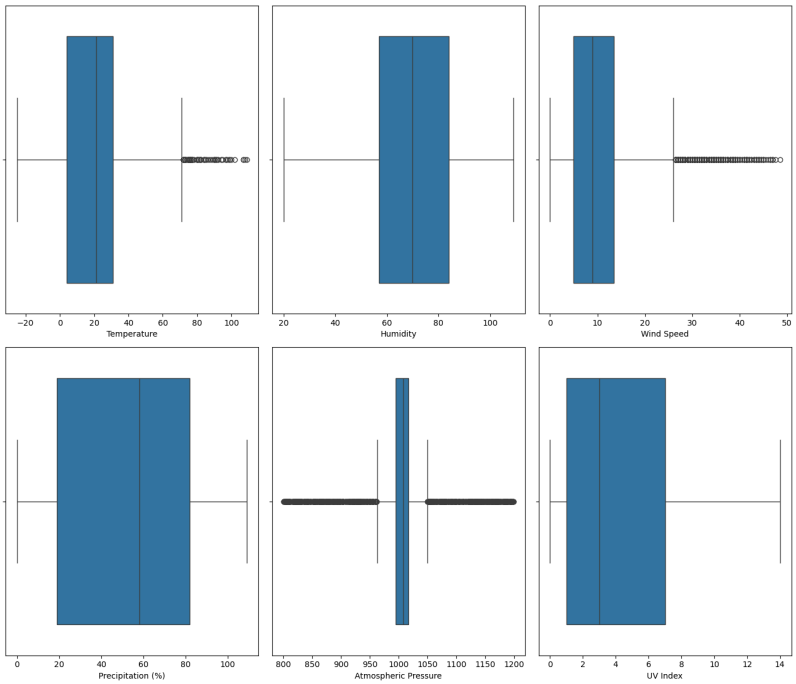
Categorical Attribute Counts:



Observations:

Varying counts in categorical attributes. However, weather type has even counts across each category. Winter is much more represented than any other season.

Numerical Attribute Boxplots:



Observations:

Temperature, Wind Speed, and Atmospheric Pressure appear to have a significant number of outliers within the dataset. However, the dataset description states that outliers have been intentionally included. Given this, the following analyses will be completed on the standard dataset as well as the dataset with removed outliers.

Outlier Removal:

For the cleansed dataset, outliers have been defined as any value beyond +/- 1.5 times the interquartile range within the numerical attributes. These values have been removed.

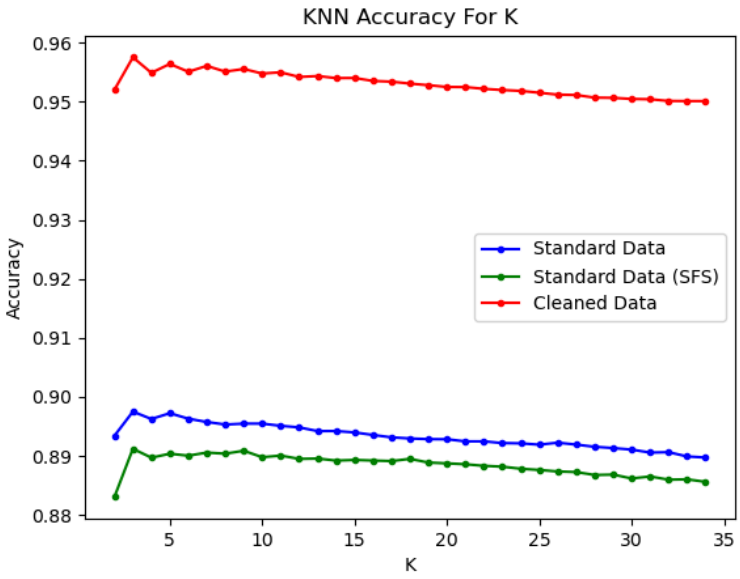
Weather Classification Using k-Nearest Neighbors

Model Reasoning:

I chose k-Nearest Neighbors (KNN) for this classification problem as it is simple and has the ability to handle multi-class problems with minimal training. KNN makes predictions based on the similarity of data points. This works well with weather data as it often exhibits clear and known patterns. However, KNN can be particularly sensitive to outlying data points and blended classes, making it useful to illustrate the importance of proper data preparation and model tuning.

Initial Results:

Standard Dataset	Cleaned Dataset
Optimal K Neighbors: 3	Optimal K Neighbors: 3
Accuracy: 89.75%	Accuracy: 95.75%



Results:
Initial implementation of the model yielded an accuracy of 89.75% with k = 3 for the standard dataset and an accuracy of 95.75% with k = 3 for the cleansed dataset. In an attempt to boost the standard accuracy scores, I used sequential feature selection with k = 3 to determine the optimal subset of features for classification.

SFS features: 'Temperature', 'Precipitation (%)', 'Atmospheric Pressure', 'UV Index', 'Visibility (km)', 'Cloud Cover Labels

SFS did not improve the accuracy of the model as it resulted with a maximum accuracy of 89.12%. Given the simplicity of KNN, it is possible that this dataset may require a more complex model or a model that is able to better capture the complex relationships between attributes while still using outlying data.

Outlier removal significantly improved the accuracy of KNN and outlined its sensitivities.

Weather Classification Using Support Vector Machine

In an attempt to boost classification accuracy, I used a Support Vector Machine (SVM). I chose this model because SVM's are effective in high-dimensional spaces (like the current weather dataset), are resistant to overfitting through proper tuning mechanisms, and its different kernel functions can be used to model different complex relationships.

Initial Results:

Standard Dataset	Cleansed Dataset
90.73%	96.69%

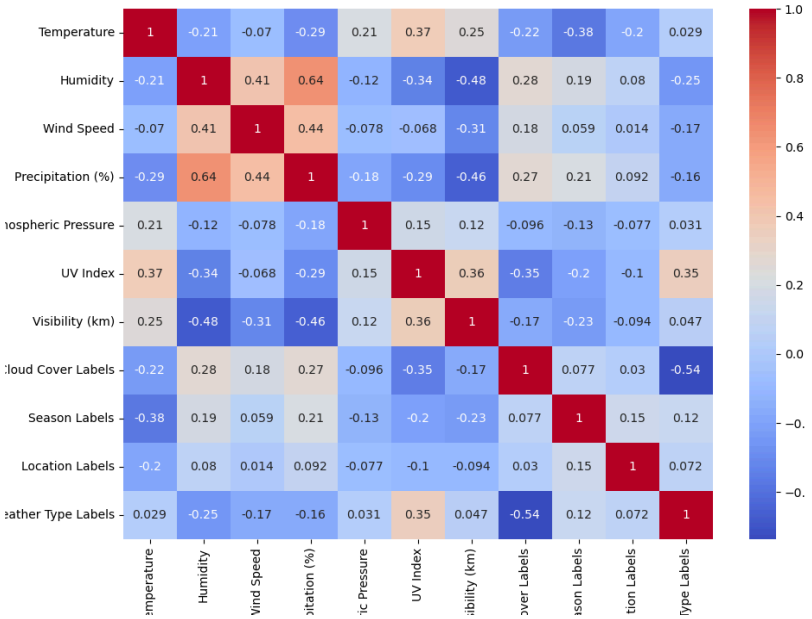
Initial implementation of SVM using a radial basis function (rbf) kernel already shows improvement when compared to KNN. To boost accuracy, hyperparameters were tuned using a grid-search cross-validation function.

Grid Search Results:

Standard Dataset	Cleansed Dataset
90.73%	96.80%
'C': 1	‘C’: 100
'break_ties': False	‘break_ties’: False
'class_weight': 'balanced'	‘class_weight’: None
'gamma': 0.1	‘gamma’: 0.01

The standard model gains no accuracy increase by tuning the hyperparameters. This suggests that the standard model performs optimally with the 'default' settings. However, tuning hyperparameters on the cleaned dataset improved accuracy slightly to 96.80% from 96.69%. The higher C value of 100 indicates a preference for classifying training examples correctly, even if the decision boundary becomes less smooth. The lower gamma value of 0.01 implies a broader reach of influence for each training example, leading to a smoother decision boundary and balancing the high C value. This configuration enhances the model's ability to generalize the cleansed data better.

Again, I used SFS to try to improve model accuracy. However, SFS did not improve model accuracy for either datasets. However, Humidity was removed from both feature spaces and the accuracy of both models with tuned hyperparameters remained the same. This could indicate that humidity has a higher degree of correlation to other attributes within the dataset.



	Feature	VIF
0	Temperature	1.405072
1	Humidity	1.943444
2	Wind Speed	1.340761
3	Precipitation (%)	1.980764
4	Atmospheric Pressure	1.069134
5	UV Index	1.436985
6	Visibility (km)	1.492405
7	Cloud Cover Labels	1.211957
8	Season Labels	1.212139
9	Location Labels	1.050672

Humidity has the highest correlation values of the attributes with Wind Speed (0.41) and Precipitation (%) (0.64). The relationship between Humidity and Precipitation is stronger, however, this relationship might be expected as humid conditions often precede or accompany precipitation.

Exploring VIF data does not lead to any significant findings as all of the values are not particularly high.

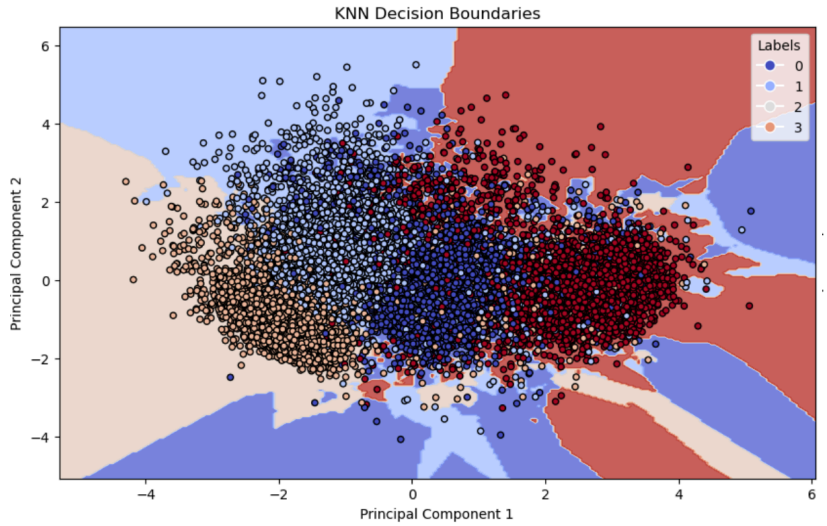
While neither of these findings are highly indicative of multicollinearity within the dataset, removing Humidity could be improving model performance through noise reduction.

KNN And SVM Decision Boundaries

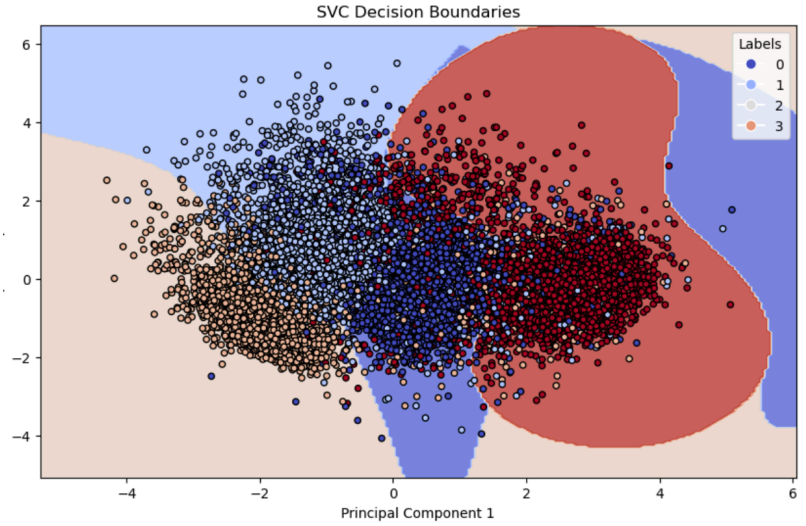
To get a better understanding of classification results, Primary Component Analysis, with 2 components, was performed on each dataset for 2D visualization of decision boundaries.

Standard Dataset

KNN

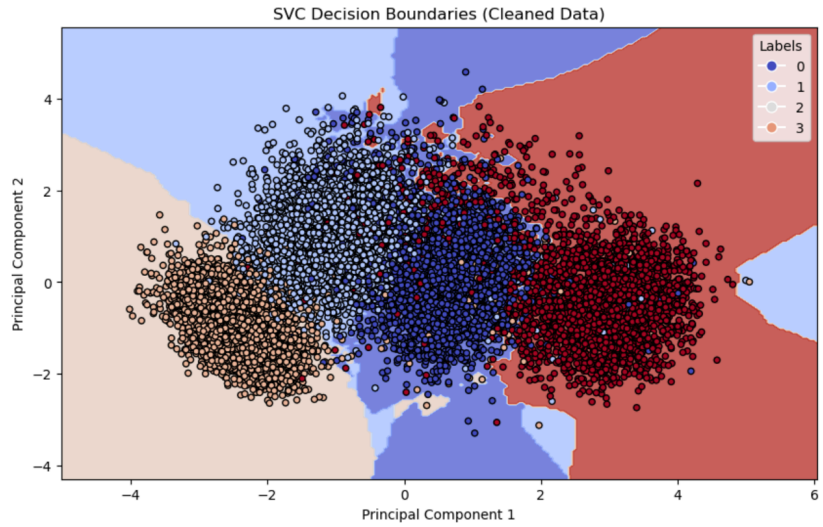


SVM

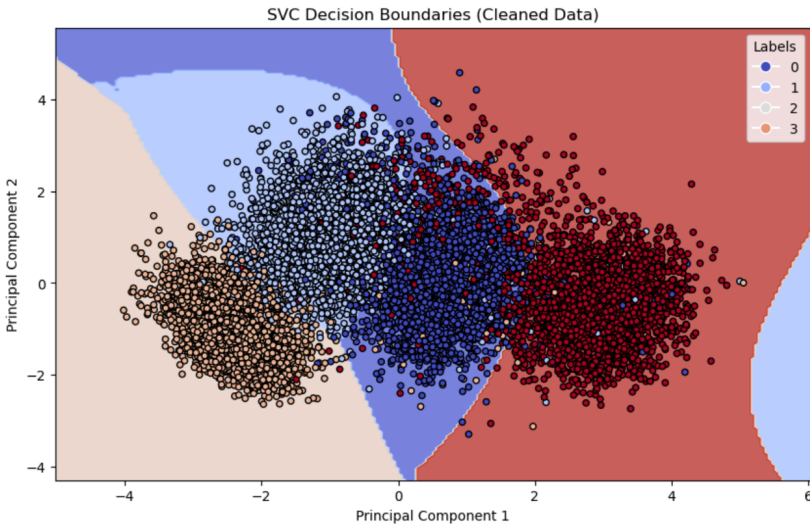


Cleansed Dataset

KNN



SVM



Observations:

When comparing the decision boundaries of KNN and SVM, it is shown that KNNs boundaries are much more chaotic and rough. SVM on the other hand is more smooth and generally more predictable. Additionally, outlier removal significantly improves boundaries.

KNN's decision boundaries are often more chaotic and rough because it makes predictions based on the closest training samples without any assumptions about the underlying data distribution. This characteristic can make KNN more sensitive to noise and outliers (as shown throughout the analysis), leading to less stable predictions, especially in complex or high-dimensional spaces.

Alternatively, SVM constructs smooth and more predictable decision boundaries by finding the hyperplanes to maximize the margin between classes. This approach allows SVM to generalize data better, especially when the data is not linearly separable. Understanding these differences is important factor when selecting appropriate models based on the characteristics of the dataset and the desired trade-off between flexibility and generalization in predictive modeling.

Summary

Historically, massive amounts of computing power has been needed to analyze and create accurate models of weather data. This analysis shows how machine learning algorithms can be used on commercial/personal hardware to achieve reasonable prediction results given proper methods and data preparation. However, this dataset begins to push the boundaries of manageable time constraints as total runtime was about 4 - 5 hours, given 11 attributes and roughly 13,000 observations. While KNN has a more accomodating time complexity of O(nd), where n is the number of samples and d is the number of attributes, SFS and and grid-search cross-validation can be quite time consuming. Additional precautions should be taken with non-linear SVMs (polynomial and rbf) as they have a time complexity between O(n^2) and O(n^3).

To explore weather classification further, additional comparisons between models like ensemble methods or neural networks could be done. Along with this, further analysis of real-world datasets could provide more insights into patterns/trends wich could improve model generalizability. Such explorations could significantly contribute to more accurate and reliable weather classification and forecasting.