Climate Change Data Analysis Using Big Data

Presented By
Krishna Teja Reddy Suram
LakshmiPriya Palaparthy

TABLE OF CONTENTS

01 Introduction

02 Business Problem

03 Data Sources

04 Data Understanding

05 Data Visualization

06 Model Building

01

Introduction

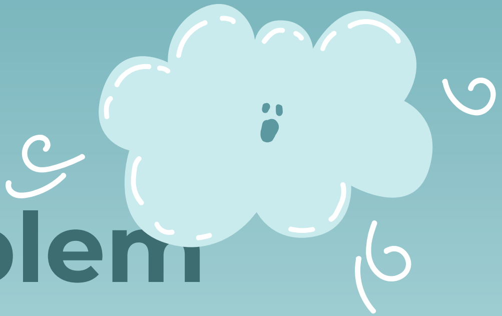
INTRODUCTION

Project Significance: Understanding weather patterns and severe weather events is crucial in today's climate-conscious society. This project plays a vital role in enhancing climate resilience by analyzing large-scale U.S. weather data to identify patterns that help predict extreme events and support disaster preparedness. Leveraging big data tools like Apache Spark on Databricks, the project provides crucial insights for data-driven climate research and mitigation strategies, aiding communities and policymakers in making informed decisions to address and adapt to changing weather patterns.

Methodology Overview: The project employs PySpark on Databricks for handling the large-scale dataset and performing time-series analysis. Key methodologies include data preprocessing, missing data handling, and temporal-spatial analysis to discover significant weather patterns. Machine learning models may be incorporated for event prediction based on historical trends

02

Business Problem



Business Problem

Problem statement:

The project aims to generate **actionable insights** into weather trends by harnessing big data analytics, helping to **identify patterns** and **predict extreme events**. This analysis serves as a valuable resource for researchers, meteorologists, and policymakers, supporting data-driven decisions that enhance climate resilience and community preparedness in the face of evolving weather patterns. write this as the business problem

Some of the challenges faced by the people of United States:

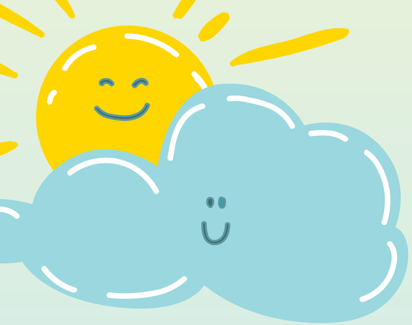
- Unpredictable extreme weather events (e.g., hurricanes, heatwaves) are becoming more frequent and intense.
- Many communities lack timely warnings and localized weather information, leaving them unprepared.
- Climate change is exacerbating these risks, making it harder for citizens to protect themselves and their property.

03

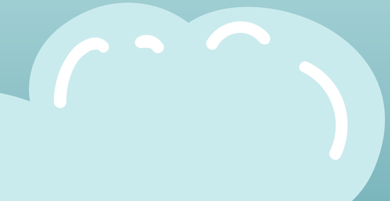
Data Source

Data Source

- **Dataset Scope**: Covers weather events across 49 states in the United States.
- **Number of Events**: Contains 8.6 million weather events.
- **Weather Types**: Includes both regular weather occurrences (rain, snow) and extreme weather events (storms, freezing conditions).
- **Time Span**: Data spans from January 2006 to December 2016.
- **Data Sources**: Sourced from 2,071 airport-based weather stations nationwide.
- **Dataset Size:** 1.1GB



04 Data Understanding

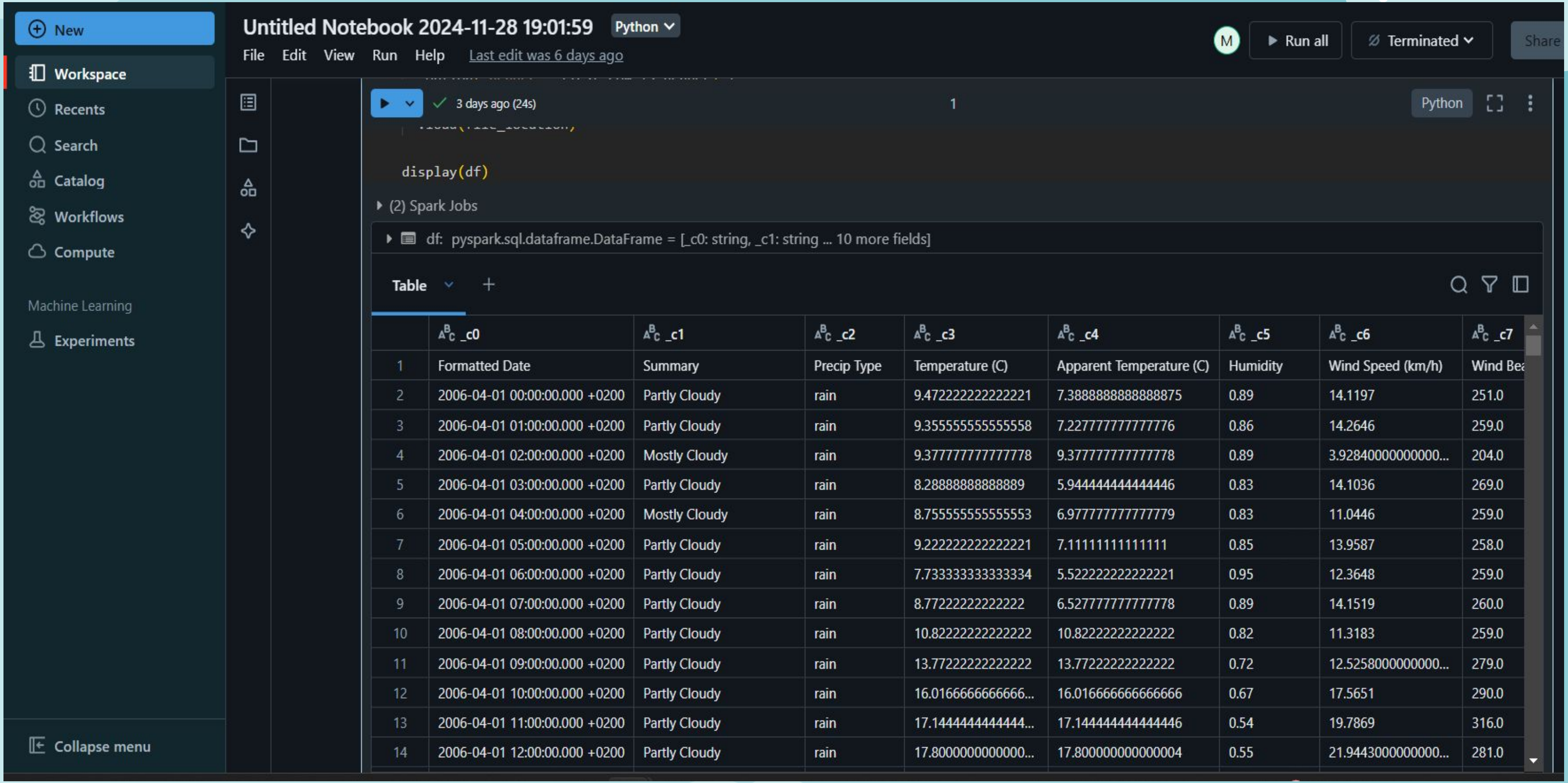


Key Variables in the Dataset

Some of the Key Variables included in the Dataset are:

1. **Formatted Date:** A timestamp with timezone information.
2. **Summary:** A brief description of the weather conditions.
3. **Precip Type:** Type of precipitation (e.g., rain).
4. **Temperature (C):** Actual temperature in Celsius.
5. **Apparent Temperature (C):** "Feels like" temperature in Celsius.
6. **Humidity:** Relative humidity (likely between 0 and 1).
7. **Wind Speed (km/h):** Speed of the wind.
8. **Wind Bearing (degrees):** Direction of the wind in degrees.
9. **Visibility (km):** How far one can see in kilometers.
10. **Loud Cover:** amount of cloud coverage
11. **Pressure (millibars):** Atmospheric pressure.
12. **Daily Summary:** A detailed weather summary for the day.

Dataset Overview



```
display(df)
```

(2) Spark Jobs

df: pyspark.sql.dataframe.DataFrame = [_c0: string, _c1: string ... 10 more fields]

	_c0	_c1	_c2	_c3	_c4	_c5	_c6	_c7
1	Formatted Date	Summary	Precip Type	Temperature (C)	Apparent Temperature (C)	Humidity	Wind Speed (km/h)	Wind Bea
2	2006-04-01 00:00:00.000 +0200	Partly Cloudy	rain	9.472222222222221	7.388888888888875	0.89	14.1197	251.0
3	2006-04-01 01:00:00.000 +0200	Partly Cloudy	rain	9.355555555555558	7.227777777777776	0.86	14.2646	259.0
4	2006-04-01 02:00:00.000 +0200	Mostly Cloudy	rain	9.377777777777778	9.377777777777778	0.89	3.92840000000000...	204.0
5	2006-04-01 03:00:00.000 +0200	Partly Cloudy	rain	8.288888888888889	5.944444444444446	0.83	14.1036	269.0
6	2006-04-01 04:00:00.000 +0200	Mostly Cloudy	rain	8.755555555555553	6.977777777777779	0.83	11.0446	259.0
7	2006-04-01 05:00:00.000 +0200	Partly Cloudy	rain	9.222222222222221	7.111111111111111	0.85	13.9587	258.0
8	2006-04-01 06:00:00.000 +0200	Partly Cloudy	rain	7.733333333333334	5.522222222222221	0.95	12.3648	259.0
9	2006-04-01 07:00:00.000 +0200	Partly Cloudy	rain	8.772222222222222	6.527777777777778	0.89	14.1519	260.0
10	2006-04-01 08:00:00.000 +0200	Partly Cloudy	rain	10.822222222222222	10.822222222222222	0.82	11.3183	259.0
11	2006-04-01 09:00:00.000 +0200	Partly Cloudy	rain	13.772222222222222	13.772222222222222	0.72	12.52580000000000...	279.0
12	2006-04-01 10:00:00.000 +0200	Partly Cloudy	rain	16.0166666666666...	16.016666666666666	0.67	17.5651	290.0
13	2006-04-01 11:00:00.000 +0200	Partly Cloudy	rain	17.1444444444444...	17.144444444444446	0.54	19.7869	316.0
14	2006-04-01 12:00:00.000 +0200	Partly Cloudy	rain	17.8000000000000...	17.800000000000004	0.55	21.94430000000000...	281.0

Dataset Cleaning

Date Formatting

Null Value Removal

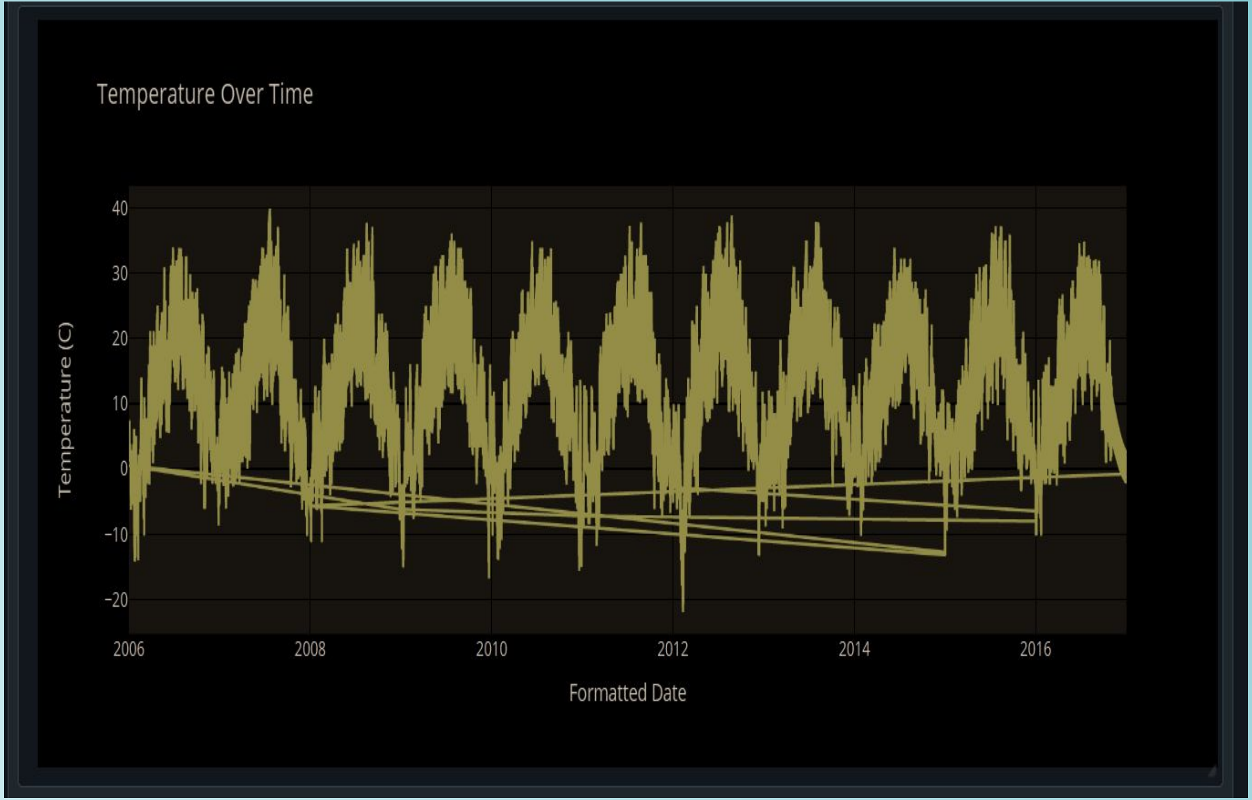
Duplicate Removal

Duplicate Removal:
Identified and removed all duplicate rows in the dataset to ensure each data point is unique.
This step prevents bias or distortion in model training and analysis.
Null Value Handling:
Addressed missing or null values by removing them from the dataset.
Ensured that the models are trained and tested only on valid, complete data points to improve reliability and accuracy.
Date Formatting:
Standardized and properly formatted the date field to maintain consistency across the dataset.
Enabled accurate handling and analysis of time-based trends and patterns.

05

Data Visualization

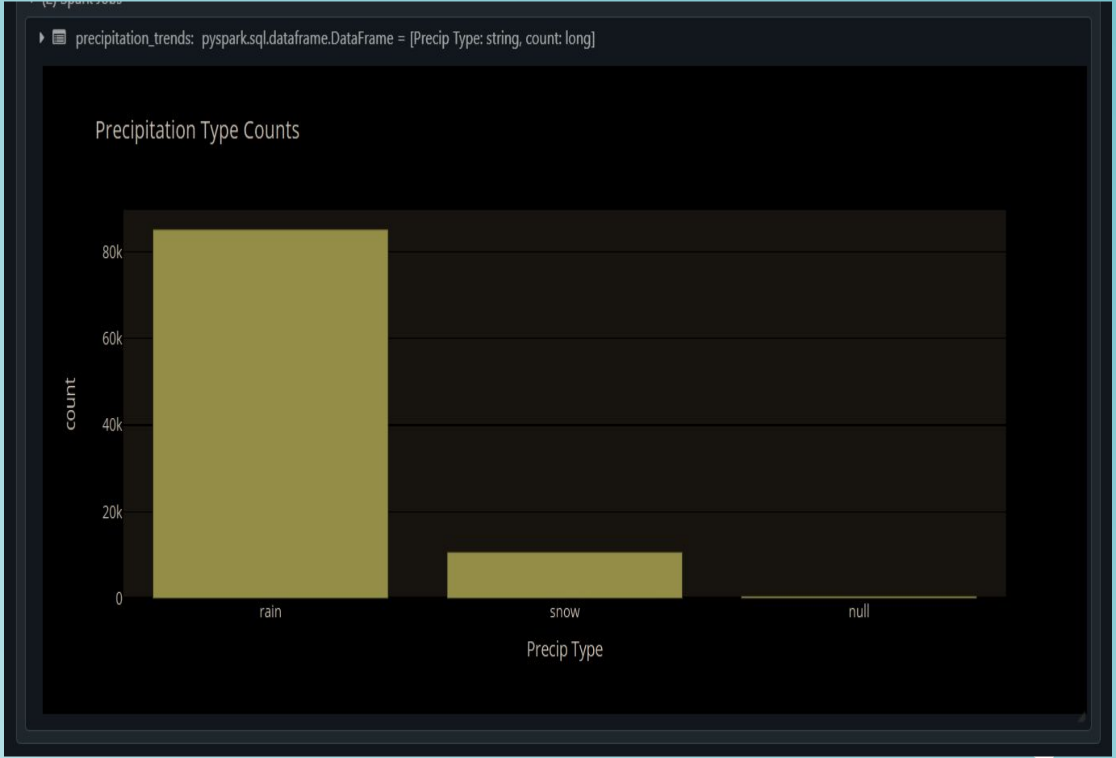
Temperature Over Time



The graph shows temperature trends over time from 2006 to 2016. The cyclical patterns indicate clear seasonal variations, with peaks during warmer months and dips during colder months. The consistency in the pattern suggests stable yearly weather cycles without significant long-term temperature changes.

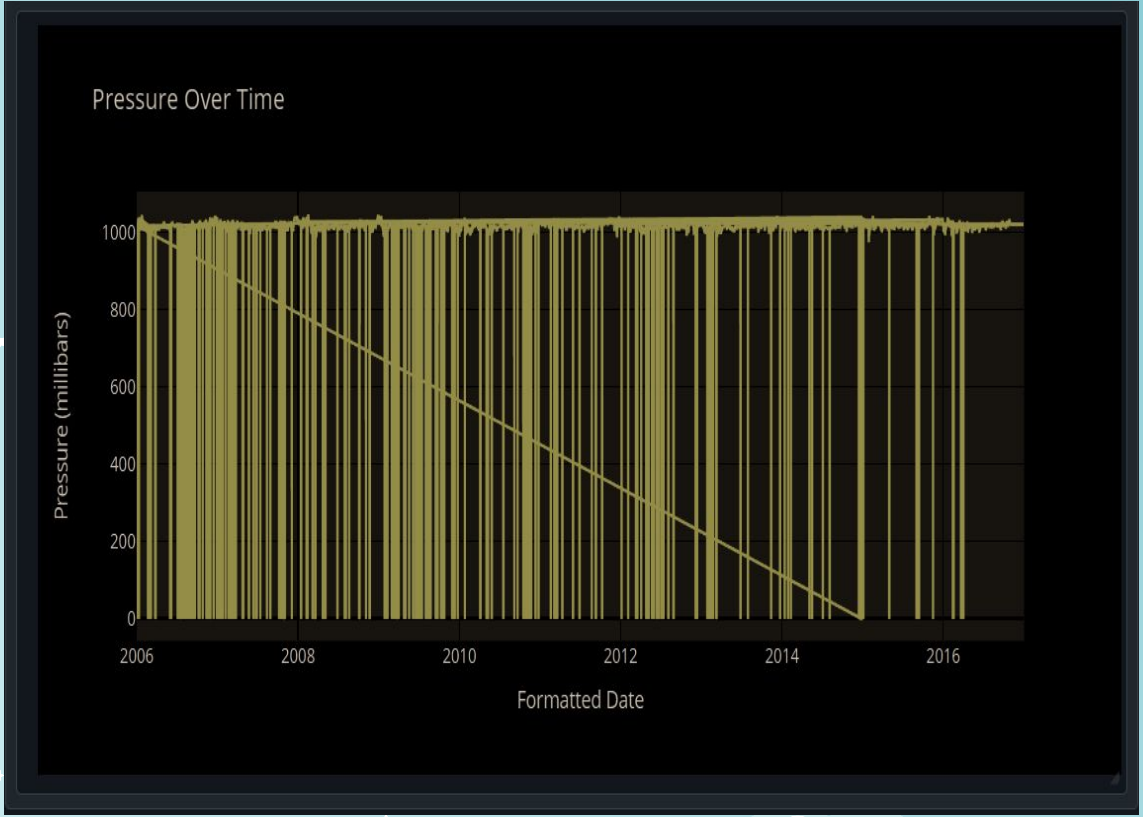
Precipitation Counts

The bar chart shows the distribution of precipitation types. Rain is the most frequent type, with over 85,000 occurrences, followed by snow, which is significantly less common with around 10,700 occurrences. The null values are negligible, indicating minimal missing data in this category. This highlights a strong dominance of rain events in the dataset.



```
precipitation_trends: pyspark.sql.dataframe.DataFrame = [Precip Type: string, count: long]
```

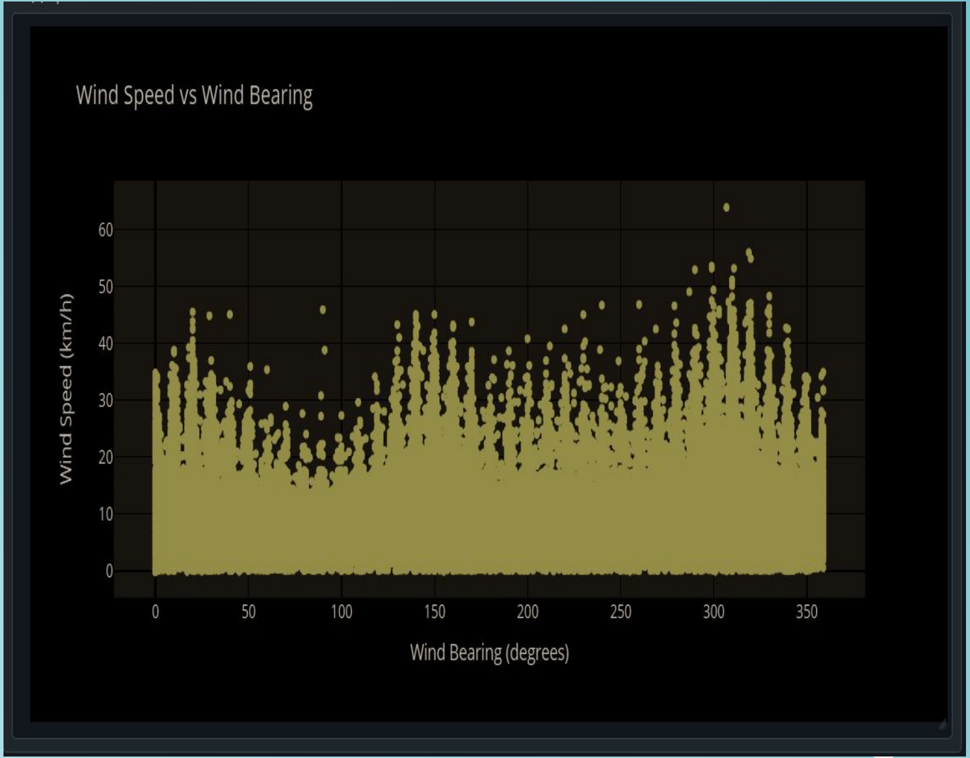
Pressure Over Time



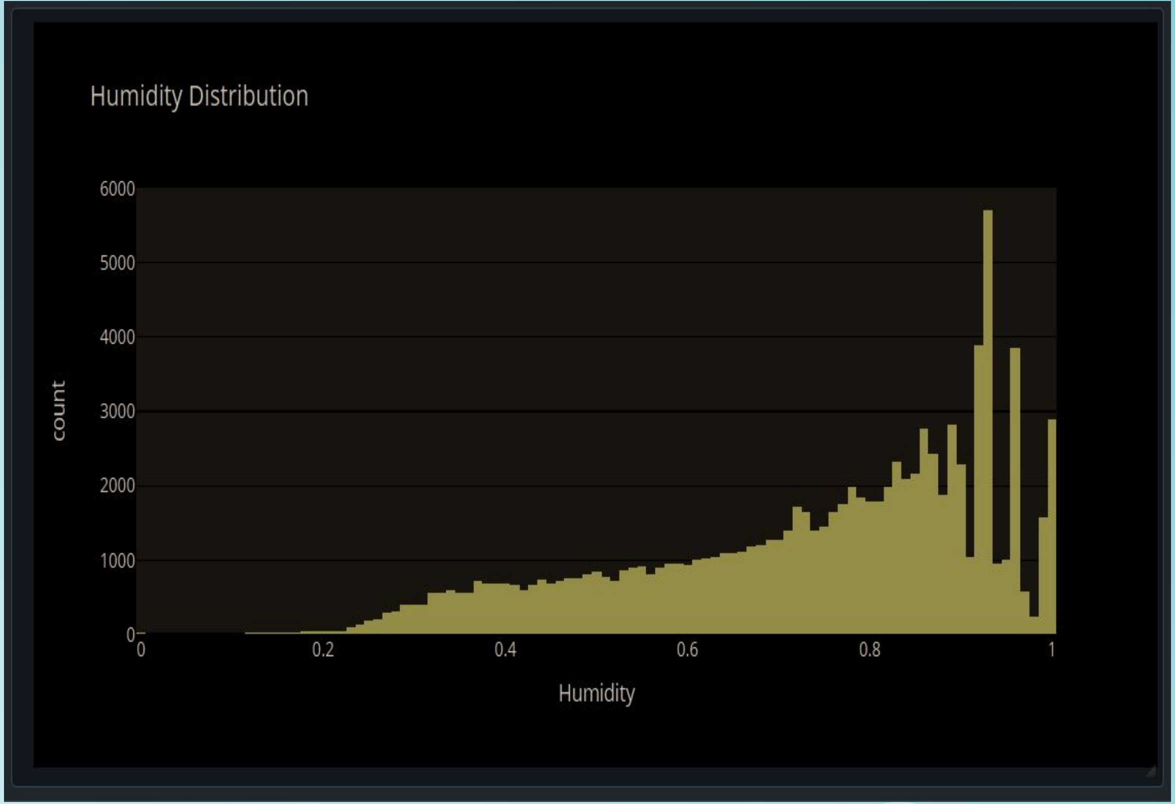
The graph shows pressure variations over time from 2006 to 2016. While most pressure readings remain relatively stable around 1000 millibars, there are significant outliers and abrupt drops, indicating possible errors or anomalies in the data collection process. These anomalies should be further investigated to ensure data quality.

Wind Speed Vs Wind Bearing

The scatter plot visualizes the relationship between wind speed (km/h) and wind bearing (degrees). Most wind speeds are below 20 km/h, with a few outliers reaching up to 60 km/h. The data appears uniformly distributed across wind bearings, indicating no specific directional bias for high wind speeds.



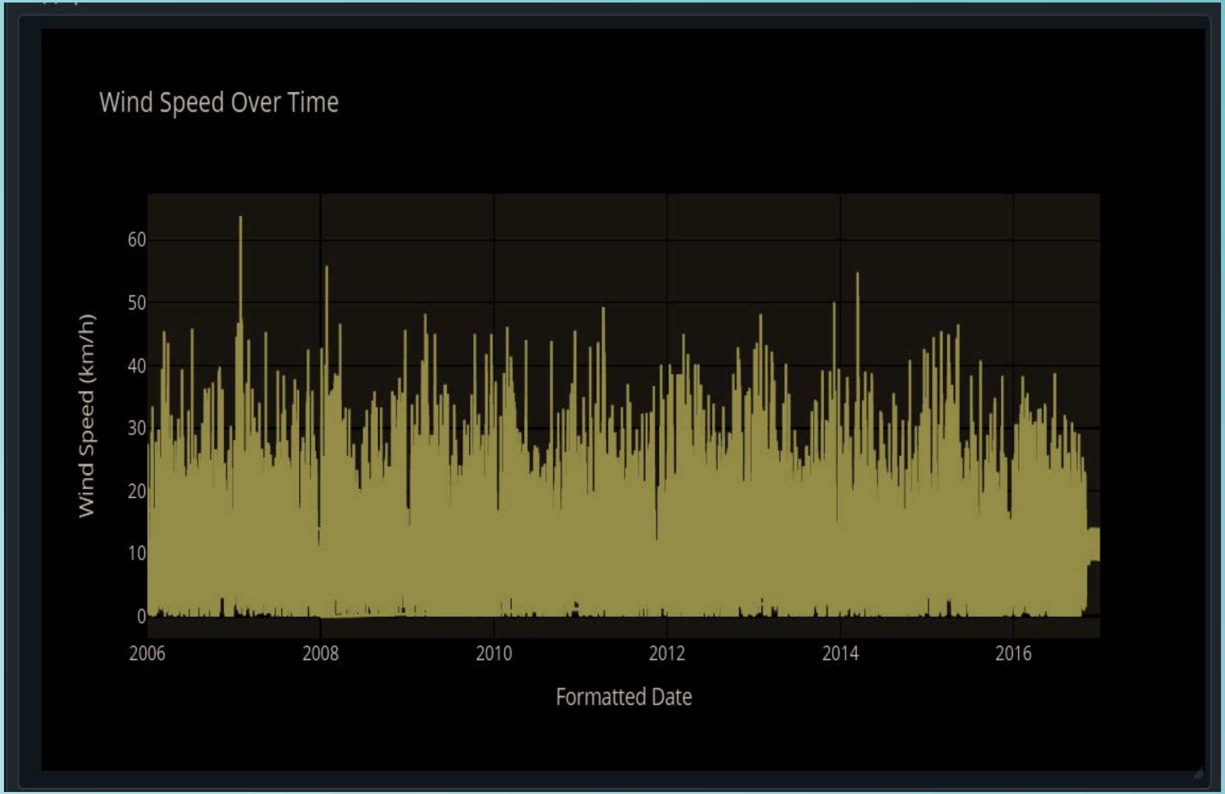
Humidity Distribution



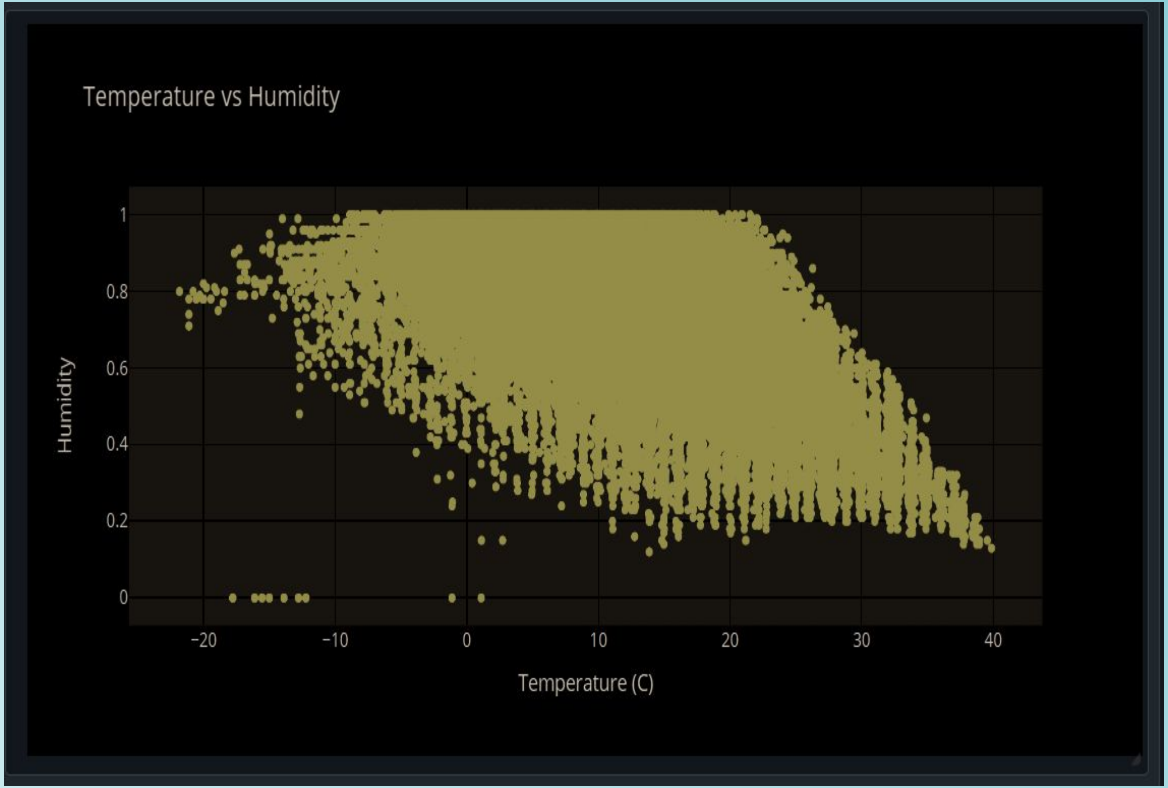
The histogram illustrates the distribution of humidity levels. Most observations have high humidity values, clustering between 0.8 and 1, indicating a predominance of humid conditions in the dataset. Lower humidity values are less frequent, with a gradual increase as humidity approaches 1.

Wind speed over Time

The graph shows wind speed trends over time from 2006 to 2016. Most wind speeds are below 20 km/h, with occasional spikes reaching up to 60 km/h. The data indicates consistent variability in wind speeds over the years, suggesting occasional high-wind events but a predominance of calmer conditions.



Temperature Vs Humidity



The scatter plot illustrates the relationship between temperature (°C) and humidity. There is an inverse relationship: higher temperatures tend to have lower humidity, while lower temperatures are associated with higher humidity levels. This aligns with common weather patterns where colder conditions retain more moisture.

Model Building

06

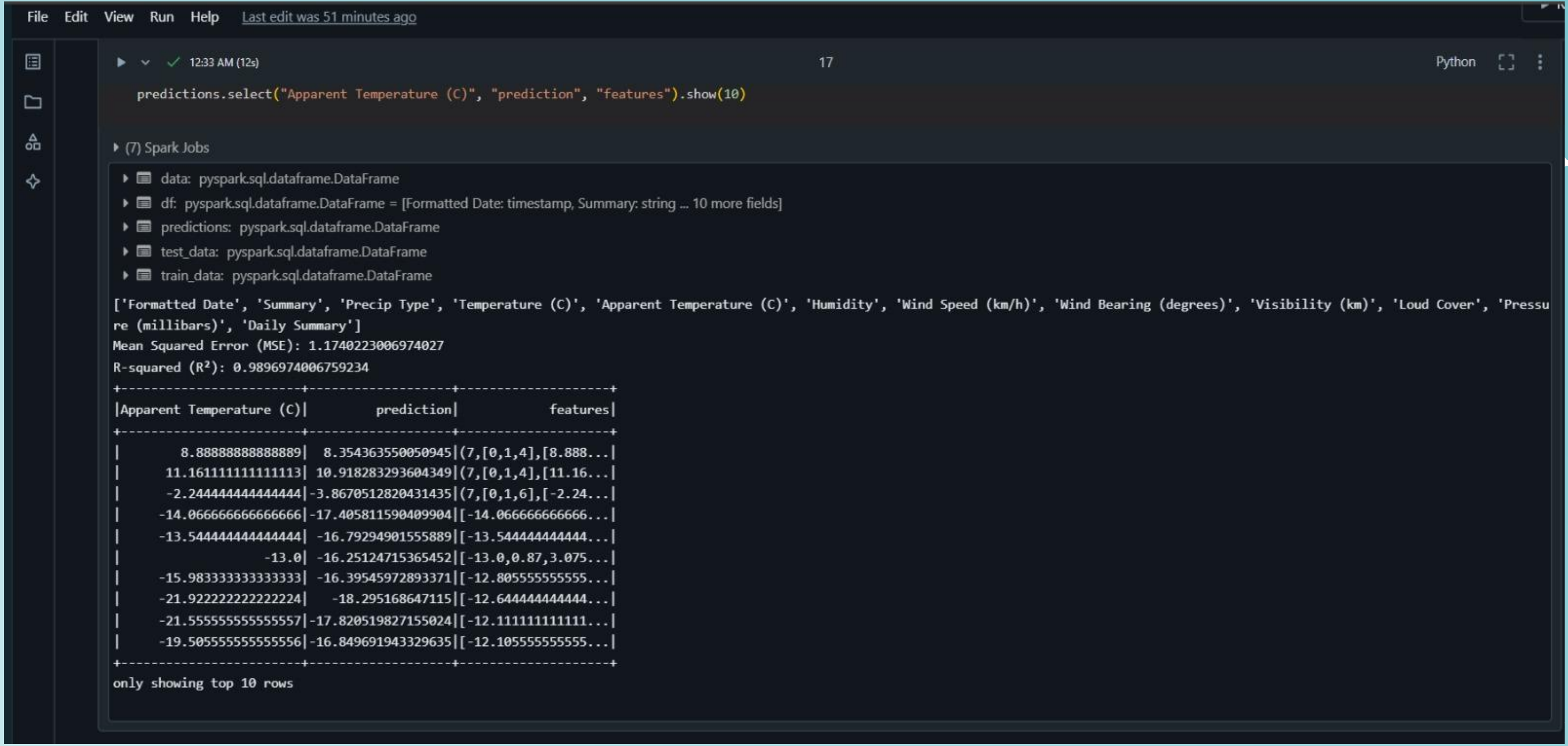
Linear Regression in PySpark

The regression model built using PySpark to predict Apparent Temperature (C) based on weather features performed remarkably well. Here's a detailed analysis of the results:

Key Metrics:

- Mean Squared Error (MSE): 1.174. This Indicates the average squared difference between the actual and predicted apparent temperatures.
- Since this value is relatively small (given the scale of temperatures in the dataset), the model predictions are highly accurate.
- R-squared (R^2):
- 0.9897: Suggests that 98.97% of the variance in the apparent temperature is explained by the features included in the model.

This indicates a very strong relationship between the features (e.g., Temperature (C), Humidity, etc.) and the target variable (Apparent Temperature (C)).



```python
predictions.select("Apparent Temperature (C)", "prediction", "features").show(10)
```

(7) Spark Jobs

- data: pyspark.sql.dataframe.DataFrame
- df: pyspark.sql.dataframe.DataFrame = [Formatted Date: timestamp, Summary: string ... 10 more fields]
- predictions: pyspark.sql.dataframe.DataFrame
- test_data: pyspark.sql.dataframe.DataFrame
- train_data: pyspark.sql.dataframe.DataFrame

```
['Formatted Date', 'Summary', 'Precip Type', 'Temperature (C)', 'Apparent Temperature (C)', 'Humidity', 'Wind Speed (km/h)', 'Wind Bearing (degrees)', 'Visibility (km)', 'Loud Cover', 'Pressu
re (millibars)', 'Daily Summary']
Mean Squared Error (MSE): 1.1740223006974027
R-squared (R²): 0.9896974006759234
+------------------------+-------------------+--------------------+
|Apparent Temperature (C)|         prediction|            features|
+------------------------+-------------------+--------------------+
|       8.888888888888889|  8.354363550050945|(7,[0,1,4],[8.888...|
|      11.161111111111113| 10.918283293604349|(7,[0,1,4],[11.16...|
|      -2.244444444444444|-3.8670512820431435|(7,[0,1,6],[-2.24...|
|     -14.066666666666666|-17.405811590409904|[-14.066666666666...|
|     -13.544444444444444| -16.79294901555889|[-13.544444444444...|
|                   -13.0| -16.25124715365452|[-13.0,0.87,3.075...|
|     -15.983333333333333| -16.39545972893371|[-12.805555555555...|
|     -21.922222222222224|   -18.295168647115|[-12.644444444444...|
|     -21.555555555555557|-17.820519827155024|[-12.111111111111...|
|     -19.505555555555556|-16.849691943329635|[-12.105555555555...|
+------------------------+-------------------+--------------------+
only showing top 10 rows
```

Random forest Classification Model

- A classification model was evaluated to predict using Random Forest model (rain, snow, or none). Below are the detailed performance metrics:

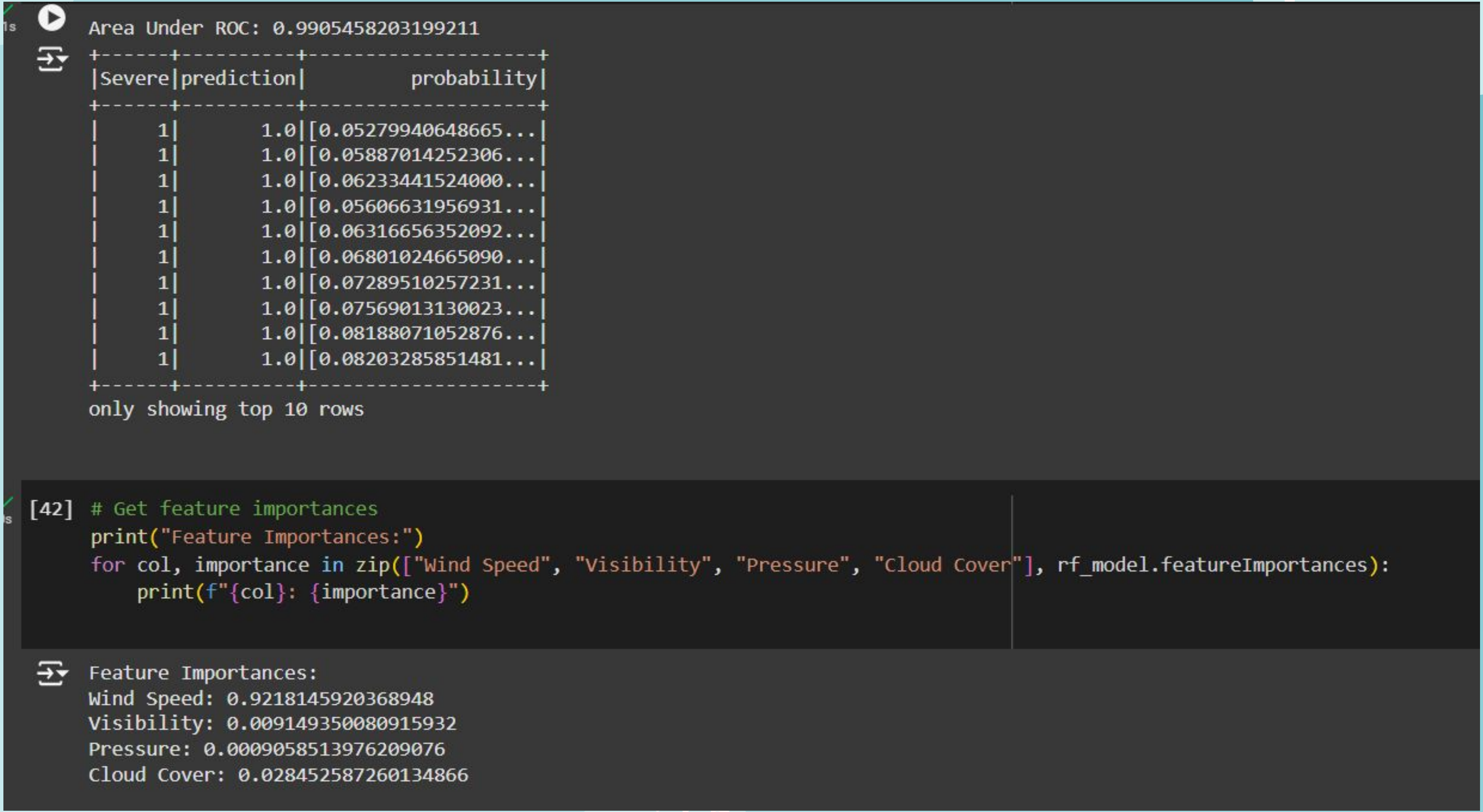
Key Metrics

- Precision: 99.51%
  Indicates the proportion of true positive predictions among all predicted positive cases. The high value demonstrates the model's accuracy in predicting precipitation types.
- Recall: 99.59%
  Represents the proportion of actual precipitation types correctly identified by the model, showcasing its ability to capture the true occurrences effectively.
- F1-Score: 99.48%
  Balances precision and recall, indicating the overall effectiveness of the model in predicting precipitation categories.
- The model showed exceptional performance, confirming its reliability for forecasting precipitation types with precision and recall across different scenarios.

Feature Importances:

Wind Speed: 0.9218145920368948
Visibility: 0.009149350080915932
Pressure: 0.0009058513976209076
Cloud Cover: 0.028452587260134866

Wind Speed is the dominant factor in determining severe weather, making it a critical variable for predicting conditions like rain or snow followed by Cloud cover.



```
Area Under ROC: 0.9905458203199211
+------+----------+--------------------+
|Severe|prediction|         probability|
+------+----------+--------------------+
|     1|       1.0|[0.05279940648665...|
|     1|       1.0|[0.05887014252306...|
|     1|       1.0|[0.06233441524000...|
|     1|       1.0|[0.05606631956931...|
|     1|       1.0|[0.06316656352092...|
|     1|       1.0|[0.06801024665090...|
|     1|       1.0|[0.07289510257231...|
|     1|       1.0|[0.07569013130023...|
|     1|       1.0|[0.08188071052876...|
|     1|       1.0|[0.08203285851481...|
+------+----------+--------------------+
only showing top 10 rows
```

```python
# Get feature importances
print("Feature Importances:")
for col, importance in zip(["Wind Speed", "Visibility", "Pressure", "Cloud Cover"], rf_model.featureImportances):
    print(f"{col}: {importance}")
```

```
Feature Importances:
Wind Speed: 0.9218145920368948
Visibility: 0.009149350080915932
Pressure: 0.0009058513976209076
Cloud Cover: 0.028452587260134866
```

Logistic Regression

- Logistic Regression is used to classify whether a weather event is severe or not (binary outcome: Severe = 1, Not Severe = 0). This simplifies the problem of predicting severe weather events based on the available weather parameters (e.g., temperature, humidity, wind speed).

- It not only classifies the event but also provides the probability of it being severe. This probability is valuable in determining how likely a given weather event is to escalate into a severe condition, allowing better preparedness and decision-making.

- This model can be deployed in real-time weather prediction systems. By continuously receiving new data (e.g., hourly or daily weather parameters), it can predict the likelihood of severe weather in real time, enabling quicker responses and actions.

Logistic Regression

The model with the following performance metrics:

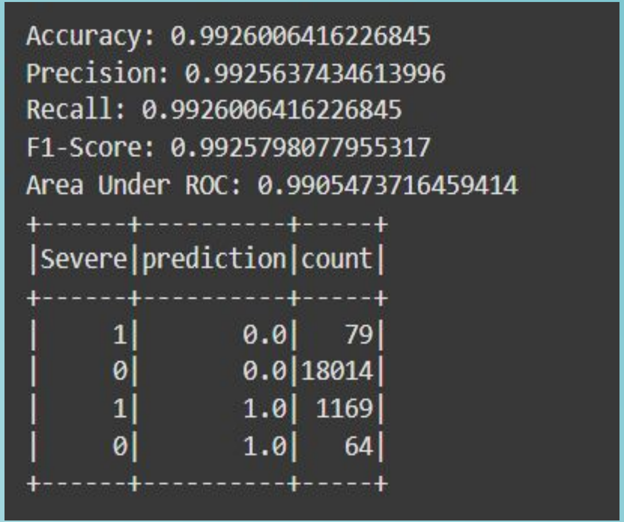
- **Accuracy:** 0.9926
- **Precision:** 0.9926
- **Recall:** 0.9926
- **F1-Score:** 0.9926
- **Area Under ROC:** 0.9905

demonstrates excellent predictive capability and can be highly suitable for analyzing the severity of climatic conditions. Its high accuracy, precision, recall, and AUC score indicate strong performance in identifying and classifying severe weather events, making it well-suited for reliable and real-time severity analysis.

Confusion Matrix

True Positives (TP): 1169 (Predicted "Severe" correctly as 1)
True Negatives (TN): 18014 (Predicted "Not Severe" correctly as 0)
False Positives (FP): 64 (Predicted "Severe" incorrectly as 1)
False Negatives (FN): 79 (Predicted "Not Severe" incorrectly as 0)

The model demonstrates excellent performance across all metrics, with minimal misclassification (only 79 false negatives and 64 false positives).
The confusion matrix shows a high number of true positives and true negatives, confirming the model’s effectiveness in classifying severe and non-severe events.



```
Accuracy: 0.9926006416226845
Precision: 0.9925637434613996
Recall: 0.9926006416226845
F1-Score: 0.9925798077955317
Area Under ROC: 0.9905473716459414
+------+----------+-----+
|Severe|prediction|count|
+------+----------+-----+
|     1|       0.0|   79|
|     0|       0.0|18014|
|     1|       1.0| 1169|
|     0|       1.0|   64|
+------+----------+-----+
```

Conclusion

In this project, we used **Databricks** to analyze a large weather dataset and extract insights through visualizations and machine learning models:

1. **Regression Model for Apparent Temperature**: Predicted apparent temperature with high accuracy, providing reliable temperature insights.
2. **Random Forest Model for Precipitation Type**: Classified precipitation types (rain, snow, none) with high precision and recall for accurate forecasting.
3. **Logistic Regression Model for Severe Weather**: Identified severe weather events with excellent performance across all metrics, offering key insights into severe weather patterns.

These models, powered by Databricks, help understand and predict weather trends, supporting better decision-making for weather forecasting and related applications.

Future work

Real-Time Prediction:

- Integrate the models into a real-time forecasting system for live weather predictions.
- Use streaming frameworks like Apache Kafka with Spark for continuous data ingestion and processing.

Expanding the Dataset:

- Extend the dataset to include global weather data to analyze broader trends and improve model generalization.
- Include recent data for more up-to-date predictions and analysis.

Enhancing Feature Engineering:

- Incorporate additional features, such as seasonal trends or geographical data, to improve the predictive power of the models.
- Experiment with derived features, like temperature fluctuations or advanced weather indices.

Thank You

ANY QUESTIONS?