Pre-report for Weather Classification Dataset

STRATEGIC DECISION MAKING WITH POWERBI

Prepared By:
Sai Sanjay Birwadkar
2023JULB01393

Table of Content

1. Problem Statement

To develop a predictive model that can classify the weather type based on various meteorological features, enabling improved decision-making for applications such as agriculture, travel, and disaster preparedness.

Key Challenges:

- Understanding the relationships between features and the target variable, "Weather Type."
- Handling potential categorical variables like "Season," "Cloud Cover," and "Location."
- Ensuring balanced representation across different weather types to avoid model bias.

2. Data Requirements

- Ensure no missing values in the target variable (Weather Type) and feature columns. Handle missing data appropriately.
- Avoid duplicate/inconsistent entries in categorical features (e.g., Cloud Cover, Season, Location). Validate numeric columns for realistic ranges.
- Ensure measurements are reliable, accurate, and conform to standard units (e.g., Atmospheric Pressure, UV Index).
- Verify numeric features cover diverse weather conditions. Use scaling methods if needed.
- Check for class imbalances in Weather Type. Apply balancing techniques like oversampling or class weighting if required.
- Include relevant features (e.g., Time of Day, historical weather patterns, geographical attributes like altitude).
- Provide sufficient detail in features like Cloud Cover (e.g., clear, overcast) and Location (e.g., coastal, inland).
- Ensure the dataset supports model scalability and performs well on unseen data.

3. Data Collection

- Weather stations, sensors (e.g., rain gauges, anemometers, barometers), and visibility sensors record core parameters like temperature, humidity, and wind speed.
- Satellites (e.g., NOAA, GOES) and radar systems (e.g., Doppler) measure cloud cover, precipitation, and wind patterns. LIDAR detects atmospheric particles.
- Real-time and historical data from platforms like OpenWeatherMap, NOAA API, or government agencies (e.g., IMD, NWS).
- Provides location-based details like altitude, terrain, and proximity to water bodies.
- Human observations for qualitative data (e.g., cloud cover) and historical records digitized for analysis.
- Seasonal classification and weather types calculated using thresholds in meteorological parameters.
- Combines historical archives, real-time streaming, and multiple sources for a comprehensive dataset.

4. Data Validation

- Numeric columns (e.g., Temperature, Humidity) should be of numeric types.
- Categorical columns (e.g., Cloud Cover, Weather Type) should have consistent labels.
- No missing values should exist. Impute or remove rows with missing data.
- Ensure values are within realistic ranges (e.g., Temperature: -50°C to 60°C, Humidity: 0–100%).
- Ensure valid categories (e.g., "clear" for Cloud Cover, "Winter" for Season).
- Remove any duplicate rows.
- Detect and address outliers using IQR or box plots.
- Check class distribution for Weather Type. Apply balancing techniques if needed.
- Ensure consistent units (e.g., km for visibility, hPa for pressure).
- Verify data alignment with geographic locations and seasonal patterns.

5. Data Cleaning

- No missing values detected.

- Outliers detected in:
  - Temperature: 92
  - Wind Speed: 404
  - Visibility (km): 383
  - Atmospheric Pressure: 927
- Other columns have no outliers based on IQR.
- Valid categories:
  - Cloud Cover: ['partly cloudy', 'clear', 'overcast', 'cloudy']
  - Season: ['Winter', 'Spring', 'Summer', 'Autumn']
  - Location: ['inland', 'mountain', 'coastal']
  - Weather Type: ['Rainy', 'Cloudy', 'Sunny', 'Snowy']
- No duplicates found.
- Options: Capping/Flooring, Transformation (log/normalization), or Removal.
- Standardize/normalize numeric columns for model compatibility.
- Use One-Hot or Label Encoding for categorical variables.
- Revalidate the dataset after cleaning for errors.

6. Tools

- Python Libraries:
  - Matplotlib: Static plots (bar charts, scatter).
  - Seaborn: Detailed visualizations (heatmaps, pair plots).
  - Plotly: Interactive charts and dashboards.
  - Pandas Profiling: Automated EDA reports.
- BI Tools:
  - Power BI, Tableau, Google Data Studio for interactive reports and dashboards.
- Excel, Google Sheets for quick analysis.
- Canva for polished presentations.
- LaTeX for technical reports.
- Jupyter Notebook, Google Colab, R Markdown for code and visualization integration.
- Kaggle, GitHub, Google Drive/OneDrive for sharing datasets and reports.
- QGIS, ArcGIS, Mapbox for geospatial data and maps.

7. Dashboard

- Load Data into Power BI by importing the cleaned dataset (CSV/Excel).
- Add key metrics (e.g., average temperature, weather type).
- Use various charts: Bar, Line, Scatter, Heatmap, Map.
- Add slicers for filtering by season, location, or weather type.
- Customize visuals (colors, labels, tooltips) and add meaningful titles and descriptions.
- Enable interactions (e.g., link bar chart to map) and use drill-through for detailed exploration.

8. Storytelling

- Identify stakeholders (e.g., meteorologists, policymakers, farmers) and focus on their interests, like safety or operational planning.
- Structure Your Story
  - Contextualize the weather's impact on daily life and introduce the problem.
  - Describe the dataset (13,200 records, features like temperature, humidity) and methodology (data cleaning, visual analysis).
  - Highlight key findings like weather type distribution, feature relationships, and seasonal patterns.
  - Connect insights into actionable outcomes (e.g., disaster preparedness, farming).
- Enhance the story with pie charts, line charts, heatmaps, and maps to highlight key points.
- Provide actionable steps like investing in weather sensors or developing public apps.