UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
Predictive Analytics Lab
Semester: 5
Batch: AIML-H-B3

Topic: Storm Surge Prediction

LAB PROJECT

Predictive analytics with python

INSTRUCTOR:
Dr. Achala Shakya

GROUP MEMBERS

Member 1
Naman Jadiya
500110793
R2142221479
B.Tech CSE AIML
Batch-3-AIML-H

Member 2
Shashank Jaiswal
500109929
R2142221438
B.Tech CSE AIML
Batch-3-AIML-H

GitHub link: https://github.com/Shashank-jais/Storm-Surge-Prediction

Report Bias-Adjusted Random Forest model

Abstract

This python project aims to make precise predictions to identify the occurrence of storm surges based on the nominal and basic features with the help of manually biased random forest model. The manual added bias helps the model to balance out the storm effecting factors based on their contribution.

The use of custom biased Random Forest greatly improved the prediction confidence and accuracy when compared to traditional random forest.

Introduction

This report summarizes the analysis and predictive modelling performed on weather data. The objectives included pre-processing the dataset, training a Random Forest classifier to predict extreme weather events, and applying bias adjustments to refine prediction probabilities. The process and outcomes are detailed in the sections below.

About the Dataset

- The dataset used is from the government website (https://www.noaa.gov).
- The dataset dates to the 1st January 2000 to 1nd January 2024. (24 Years dataset Initially).
- The data maps the weathering status for the geographical region of Jacksonville Florida US.
- The reason for selecting the location of Jacksonville, primarily Florida because it is very prone to storms and cyclones which would be ideal to capture the required information to train storm prediction model and Jacksonville has the closest weather centre from the ocean.
- Parameters used are stated and defined in the appendix

Data Pre-processing

- Key pre-processing steps included:(The abbreviation used are defined below in the Appendix)
- Removing columns such as 'STATION', 'SNOW', and 'SNWD' because the coastal areas do not face snow fall. Removing “STATION” because it’s a non-numeric term which doesn’t contribute in model training.
- Filling missing values with zeros for certain features or using mode (e.g., for 'WSF5').
- Filling the missing WT* (weather condition) values with 0.
- Creating a target column, 'EXTRM', by matching specific dates to label extreme weather events. The “EXTRM” provides a form of supervised learning support which in turn converts this model to semi supervised model.
- The used values in “EXTRM” is collected from a well trusted and reliable data source (https://www.weather.gov/jax/events).
- Now removing the “DATA” column from the final dataset. So, that we don’t have to deal with (-) sign in data
- Finally, null values were checked to ensure data integrity.

Proposed Approach

The approach used depends upon manually altering the biases for all parameters based on the influence they have on any storm or cyclone. For example, let's consider a parameter used "WSF1" which defines the fastest wind in under 1 minute which upon being a larger value can contribute to a storm surge. So, it is provided with a positive bias but, a fast wind just in 1 minute can't cause a storm surge and where the other parameters come in play. If all other weather conditions are normal then the wind could have been from any external source like a the constantly low flying patrolling planes near the coastline. Other factors with negative bias also balance this positive bias in case if the wind speed was natural.

Managing uncertainties, the column EXTRM was add with the special bias "Very very high" which would lead the probability for a strong weather condition to almost 1. Considering an example of an asteroid strike near the coastline would probably create a very strong ocean current but, along with the current would also create a sudden change in many parameters used in the model leading to the overall outcome to be positive.

Modelling Process

A Random Forest Classifier was employed for the task. The dataset was split into training and testing sets (80:20 ratio). The model was trained using default hyperparameters initially, followed by optimization using GridSearchCV. Performance metrics such as accuracy and confusion matrix were used for evaluation.

The parameters obtained by the GridSearchCV method did not improve the final prediction accuracy instead reduced the confidence score of events. Visualized in code with confusion matrix.

Bias Adjustment Mechanism

Bias adjustment was applied to refine prediction probabilities. Each feature was assigned a bias category ('neutral', 'positive', 'negative', etc.).

For example:

- 'Neutral' features: Adjusted based on the mean value of the feature.
- 'Positive' features: Probabilities increased by 10%.
- 'Very high' features: Probabilities increased by 50%.

This mechanism aimed to align predictions with domain-specific knowledge as discussed in the approach section.

Results and Evaluation

The Random Forest model's performance was assessed using:

- Confusion matrix visualization.
- Accuracy, precision, recall, and F1-score.

After bias adjustment, predictions were re-evaluated, and results showed improved alignment with expected outcomes and the final accuracy comes out to be 97%.

Feature Distributions

Feature distributions were visualized to identify trends and outliers. Key observations included:

- Seasonal patterns in temperature-related features.
- Variability in precipitation and wind speed features.

Conclusion and future directions

The analysis demonstrated the effectiveness of custom biased Random Forest classifier in predicting extreme weather events. Bias adjustments further enhanced prediction reliability.

Future improvements consist

- Adding more parameters
- Providing geographical topology of the area
- Making the model location independent to more extent.
- Improving the bias balancing to avoid miss handling of small changes
- Adding ocean current patterns

Appendix

GridSearchCV was used for hyperparameter tuning. The best parameters identified were as follows:
- Number of estimators: 100.
- Maximum depth: 20.
- Minimum samples split: 2.
- Minimum samples leaf: 1.
- Max features: 'sqrt'.

Abbreviations Used

The five core values are:
PRCP = Precipitation (mm or inches as per user preference, inches to hundredths on Daily Form pdf file)
SNOW = Snowfall (mm or inches as per user preference, inches to tenths on Daily Form pdf file)
SNWD = Snow depth (mm or inches as per user preference, inches on Daily Form pdf file)
TMAX = Maximum temperature (Fahrenheit or Celsius as per user preference, Fahrenheit to tenths on Daily Form pdf file
TMIN = Minimum temperature (Fahrenheit or Celsius as per user preference, Fahrenheit to tenths on Daily Form pdf file

The other values are:
AWND = Average daily wind speed (meters per second or miles per hour as per user preference)
PGTM = Peak gust time (hours and minutes, i.e., HHMM)

WDF1 = Direction of fastest 1-minute wind (degrees)
WDF2 = Direction of fastest 2-minute wind (degrees)
WDF5 = Direction of fastest 5-second wind (degrees)
WDFG = Direction of peak wind gust (degrees)
WDFI = Direction of highest instantaneous wind (degrees)
WDFM = Fastest mile wind direction (degrees)
WDMV = 24-hour wind movement (km or miles as per user preference, miles on Daily Form pdf file)
WESD = Water equivalent of snow on the ground (inches or mm as per user preference)
WESF = Water equivalent of snowfall (inches or mm as per user preference)
WSF1 = Fastest 1-minute wind speed (miles per hour or meters per second as per user preference)
WSF2 = Fastest 2-minute wind speed (miles per hour or meters per second as per user preference)
WSF5 = Fastest 5-second wind speed (miles per hour or meters per second as per user preference)
WSFG = Peak guest wind speed (miles per hour or meters per second as per user preference)
WSFI = Highest instantaneous wind speed (miles per hour or meters per second as per user preference)
WSFM = Fastest mile wind speed (miles per hour or meters per second as per user preference)
WT** = Weather Type where ** has one of the following values:
01 = Fog, ice fog, or freezing fog (may include heavy fog)
02 = Heavy fog or heaving freezing fog (not always
distinguished from fog)
03 = Thunder
04 = Ice pellets, sleet, snow pellets, or small hail
05 = Hail (may include small hail)
06 = Glaze or rime
07 = Dust, volcanic ash, blowing dust, blowing sand, or blowing obstruction
08 = Smoke or haze
09 = Blowing or drifting snow
10 = Tornado, waterspout, or funnel cloud
11 = High or damaging winds
12 = Blowing spray
13 = Mist

14 = Drizzle
15 = Freezing drizzle
16 = Rain (may include freezing rain, drizzle, and freezing drizzle)
17 = Freezing rain
18 = Snow, snow pellets, snow grains, or ice crystals
19 = Unknown source of precipitation
21 = Ground fog
22 = Ice fog or freezing fog

WVxx = Weather in the Vicinity where "xx" has one of the following values
01 = Fog, ice fog, or freezing fog (may include heavy fog)
03 = Thunder
07 = Ash, dust, sand, or other blowing obstruction
18 = Snow or ice crystals
20 = Rain or snow shower

Files Significance

- **3819112.csv**: Contains the original dataset.
- **Label-Info.txt**: Consists of all the actual label information.
- **predicted_results.csv**: Stores the pre-processed dataset along with the added column containing predicted values of the custom bias random forest.
- **predicted_results_with_grid_search.csv**: Stores the pre-processed dataset along with the added column containing predicted values of the Gridsearch optimized custom bias random forest.
- **predicted_results-SRF.csv**: Stores the pre-processed dataset along with the added column containing predicted values of the Standard Random Forest just for comparison.
- **Predictive-Lab_Assignment.ipynb:** It is the python notebook containing the step by step processing of pre-processing, model training and evaluation.

Order of Execution

- Initialize the git bash in the "Predictive-Project" folder
- Run the command chmod +x run.sh to make the file executable.
- Run the command ./run.sh to run the script file to manage dependencies to ensure the continent execution of entire python notebook.
- Now you can execute the the entire python notebook.