	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	1 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

CECALT Hurricane Behavior Predictor

Standard Operating Procedure (SOP)

1. Purpose

This project, called CECALT Hurricane Behavior Predictor, is a process developed from start to finish with the objective of demonstrating that the developed model can improve the reliability of wind speed prediction, taking into account the peak wind speed, as well as relevant data, such as latitude, longitude, pressure, air temperature, relative humidity, precipitation, wind direction, average wind of the phenomenon and the rate of change in wind speed.

In addition, the initial motivation lies in two main purposes, the first resides in the most recent catastrophe of Hurricane Otis in Mexico and the second in improving the Saffir-Simpson hurricane wind scale.

2. Scope

CECALT Hurricane Behavior Predictor has an approach based especially on open datasets that with WebScrapping, NLP and ensemble learning methods offers a template that can be applied in international scenarios.


3. Prerequisites

Knowledge in Machine Learning and programming.

4. Responsibilities

Aarón Castillo Medina - Data Engineer and ML Engineer

- Background and Investigation
- Data Collection and Preprocessing
- Model Interpretation
- Deployment and Monitoring
- Collaboration and Communication

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	2 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

Javier Sánchez Silva - Data Scientist

- Background and Investigation
- Data Collection and Preprocessing
- Model Interpretation
- Deployment and Monitoring
- Collaboration and Communication

Gilberto Subías García - Data Scientist

- Background and Investigation
- Feature Engineering
- Model Selection and Training
- Model Evaluation
- Model Interpretation
- Collaboration and Communication


5. Procedure

Provide the steps required to perform this procedure (who, what, when, where, why, how). Include a process flowchart.

5.1 Initiation

CECALT is an analytical solution driven by machine learning techniques. Developed from inception to execution, its primary objective is to demonstrate the efficacy of ensemble learning in enhancing wind speed prediction accuracy, incorporating peak wind speed alongside supplementary data such as latitude, longitude, pressure, air temperature, relative humidity, precipitation, wind direction, average wind speed, and the rate of change in wind speed.

- Project team:
 - Aarón Castillo Medina - Data Engineer and ML Engineer
 - Javier Sánchez Silva - Data Scientist

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	3 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- Gilberto Subias García - Data Scientist

5.2 Planning and Execution

To achieve the main purpose of CECALT, the project were divided as follows:

1. Background and Investigation


Over the last century, climate change has emerged as a significant challenge, contributing to a surge in natural disasters such as droughts, severe storms, temperature fluctuations, and hurricanes.

For instance, in October 2023, a notable event unfolded in Acapulco Guerrero, Mexico. An unprecedented tropical storm rapidly escalated into a category 5 hurricane within a mere 24 hours. Caught off guard by this sudden intensification, coastal residents were unable to adequately prepare, resulting in greater devastation than anticipated and causing irreparable and mournful losses.

This highlights a critical gap in the traditional methodologies: relying solely on historical wind speed data without considering other factors that influence the development and intensity of such meteorological phenomena. Consequently, forecasts issued by various agencies in the lead-up to the event proved inaccurate, underestimating the strength of the hurricane. Had these forecasts been more precise, preemptive actions could have been taken, potentially mitigating the impact.

2. Feature Engineering

We initiated by gathering public data from the National Hurricane Center website. Employing NLP and web scraping techniques, we extracted and processed this data, augmenting it with additional variables such as air temperature, relative humidity, precipitation, wind direction, average wind speed, and the rate of change in wind speed

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	4 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

using the Meteostat library. This yielded a comprehensive dataset spanning over a decade, capturing hourly records of over 700 hurricanes.

The next code sections will explain in details as follows:

2.1 PDFs Web Scrapping

a. Purpose

This code demonstrates how to crawl a website and extract specific files based on a regular expression pattern. In this case, the code crawls the National Hurricane Center's website to download all PDF files related to tropical cyclone reports.

b. Packages Used

- ``re``: Used for regular expression matching.
- ``wget``: Used to download files from the web.
- ``scrapy``: Used to crawl the website and extract links.


c. Workflow

- i. ****Define the Spider:**** A Scrapy spider class is defined to crawl the website and extract links to PDF files.
- ii. ****Start URLs:**** The spider is configured with a list of start URLs to begin crawling.
- iii. ****Parse Function:**** The spider's ``parse`` function extracts links from each page and filters them based on the regular expression pattern.
- iv. ****Write Links to File:**** The extracted links are written to a text file on Google Drive.
- v. ****Download Files:**** The downloaded links are then downloaded using the ``wget`` library.

d. Output

The code generates a text file containing the links to all PDF files related to tropical cyclone reports found on the National Hurricane Center's website. These files are also downloaded to Google Drive.

e. Additional Notes

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	5 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- The regular expression pattern used in this notebook is `^[A-Za-z0-9]+_[A-Za-z0-9]+.pdf$``. This pattern ensures that only files with the `.pdf`` extension and a specific naming convention are downloaded.
- The ``wget`` library is used to download files from the web. This library offers various options for downloading files, such as specifying the output filename and directory.

2.2 Transform PDFs into .txt files

a. Purpose

This code demonstrates the process of converting all pdf files within a specified directory into individual text files.

b. Packages Used

It utilizes the ``os``, ``pandas``, ``re``, ``numpy``, ``time``, ``PyPDF2``, ``pdfplumber``, and ``json`` libraries.


c. Workflow:

i. ****Defining the `convert2text` Function**:**

- This function takes a ``archivo`` (filepath) as input and returns the extracted text from the PDF file.
- It uses ``pdfplumber`` to open the PDF file and extract text from each page.
- The extracted text is concatenated and returned.

ii. ****Processing all PDFs**:**

- Define the path to the directory containing the pdf files.
- ``for filename in os.listdir(path)``: Iterates over each filename in the directory.
- ``fullpath = os.path.join(path, filename)``: Constructs the full path to the current PDF file.
- ``txt_file_name = fullpath[:57]+'alltxts/' + fullpath[65:-4]+''.txt``: Creates the filepath for the corresponding text file.
- ``text_file = open(txt_file_name, 'wt')``: Opens the text file in write mode.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	6 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- ``text = convert2text(fullpath)``: Calls the ``convert2text`` function to extract text from the current PDF file.
- ``text_file.write(text)``: Writes the extracted text to the text file.
- ``text_file.close()``: Closes the text file.
- ``print(Finish with: ', filename)``: Prints a message indicating the completion of processing for the current file.

d. Output

This code successfully demonstrates the conversion of all pdf files within a specified directory into individual text files.

2.3 Extracting information from .txt files

a. Purpose

This code is designed to process and clean hurricane track data from text files and save the cleaned data as CSV files. It iterates through all text files in a specified directory, extracts relevant information, and generates a new CSV file for each text file.

b. Packages Used

It utilizes the ``os``, ``pandas``, ``re``, ``numpy`` and ``time`` libraries.

c. Workflow

i. ****File Reading:****

- Reads the text file using the ``open()`` function.
- Stores the text content in the ``txt`` variable.

ii. ****Hurricane and Period Extraction:****


- Extracts the hurricane name and period from the text using regular expressions.
- Handles cases where the format might differ.

iii. ****Information Extraction:****

- Extracts information about the hurricane track, including date, time, latitude, longitude, pressure, wind speed, and stage.
- Handles different date and time formats.

iv. ****Data Cleaning:****

- Removes unnecessary information from the data.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	7 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- Concatenates relevant fields to create the hurricane stage.
- Cleans and formats the stage field.

v. ****Date and Time Calculation:****

- Calculates the correct date and time for each data point based on the extracted information.

vi. ****Additional Fields:****

- Adds the hurricane name and a sequential ID to each data point.

vii. ****Data Formatting:****

- Selects the relevant columns for the final DataFrame.
- Converts data types for numeric columns.

viii. ****CSV Generation:****

- Saves the cleaned data as a CSV file with the appropriate filename.

d. **Additional Information:**

- The notebook keeps track of the progress and prints the percentage completed.
- It also counts the number of files processed and the number of files discarded due to errors.

2.4 Preprocessing Meteorological Information

a. **Purpose**

The aim of this code is to add Meteostat's additional data to the original PDF's tables located in .txt files, and fancy them in order to prepare them for the next stage of analysis.


b. **Packages Used**

It utilizes the ``os``, ``time``, ``math``, ``numpy``, ``pandas``, ``datetime``, ``matplotlib.pyplot`` and the ``meteostat`` libraries.

c. **Workflow**

i. ****Data Manipulation and Transformation:****

- The ``df`` DataFrame is manipulated to add a new column named ``sequential_id`` and perform various operations based on this identifier.
- The ``closest_stations`` function uses a loop to find the closest meteorological stations for each data point based on latitude and longitude.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	8 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved


- The ``meteorological_info`` function retrieves meteorological data for each data point using the closest station and the ``meteostat`` library.
- The functions use conditional statements and data manipulations to fill missing values and perform calculations.
- ii. ****Dictionary Access and Manipulation:****
 - The code uses dictionaries like ``dict_name`` and ``dict_name_reversed`` to map between names and numerical identifiers. This helps in manipulating and accessing data based on these identifiers.
- iii. ****Download File from URL:****
 - The ``wget.download`` function is used to download a file from a URL and store it locally.
- iv. ****Meteorological Data Retrieval:****
 - The ``meteostat`` library is used to fetch meteorological data for a specific location and time period. The data is stored in a DataFrame and accessed using indexing.
- v. ****Calculations and Aggregations:****
 - Loops and conditional statements are used to perform calculations and aggregations on the data. For example, the code checks for missing values and fills them with appropriate values based on available data.
- d. **Output**

This code demonstrates various data processing techniques and manipulations using Python libraries like pandas, numpy, and meteostat. It provides the final dataset for the analysis exploration stage.

3. Exploratory Data Analysis

- As the last sections, following we can find the code sections that describes in more details this part of the project:
- a. **Purpose**

This code provides a comprehensive overview of the data exploration and preprocessing steps performed on the dataset. It includes detailed descriptions of each step, along with the rationale behind the chosen methods.
 - b. **Workflow**
 - i. ****Data Import and Overview****
 - The dataset was initially imported using `pandas.read_csv()` function.
 - Basic information about the data was obtained using `df.info()` and `df.head()` methods.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	9 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- The names of the columns were changed to more descriptive ones using `df.rename()` method.
- Wind speed values were converted from kt to km/h by multiplying by 1.852.
- A new column named "rate_of_change_wind_speed" was created to calculate the rate of change of wind speed for each sequential_id.

ii. ****Missing Values****

- The total number of missing values and their percentages for each column were calculated using `df.isnull().sum()` and `df.isnull().count()` methods.
- The columns with missing values were identified and listed.

iii. ****Categorical Data****


- The categorical variables in the dataset were identified using `df.select_dtypes(include='object')` method.
- The unique values and their counts for each categorical variable were calculated using `df.groupby()` and `df.count()` methods.

iii. ****Numerical Data****

- The numerical variables in the dataset were identified using `df.select_dtypes(include=np.number)` method.
- Descriptive statistics for the numerical variables were calculated using `describe_plus()` function.
- Distribution plots were created for each numerical variable using seaborn's `distplot()` function.
- Boxplots were created for each numerical variable using seaborn's `boxplot()` function.

iv. ****Time Series Analysis****

- A datetime column was created from the `timestamp_utc` column using `pd.to_datetime()` function.
- New columns for year, month, day, and hour were extracted from the datetime column.
- The average wind speed for each day of the week for each hurricane was calculated using `df.groupby()` and `df.mean()` methods.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	10 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- The typical wind and pressure patterns during the day were visualized using bar plots.
- Time series plots were created for wind speed and pressure for each year.

v. ****Correlation Analysis****

- Pearson and Spearman correlation coefficients were calculated for all pairs of numerical variables using `df.corr()` method.
- The top 5 positive and negative correlations for each coefficient were identified and printed.
- Heatmaps were created to visualize the correlation matrices.

vi. ****Predictive Power Matrix****

- The predictive power matrix was calculated using `pps.matrix()` function from the `ppscore` library.
- The top 10 features with the highest predictive power for the target variable were identified and displayed.
- A heatmap was created to visualize the predictive power matrix.

4. Model Selection, Training and Evaluation

a. Purpose

This code provides a comprehensive analysis of a dataset containing various meteorological parameters and wind speed measurements. The primary objective is to develop a model capable of accurately predicting wind speed based on the available features.


b. Workflow

i. Data Overview

The dataset includes several features such as pressure, air temperature, dew point, relative humidity, precipitation, condition code, wind direction, average pressure, average wind speed, and rate of change of wind speed.

ii. Data Preprocessing

- * Missing values were filled with the minimum value of each respective feature.
- * Categorical features were identified and analyzed.

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	11 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

- * Numerical features were examined using descriptive statistics and visualizations.

- * Time series plots were created to analyze the wind speed and pressure patterns over time.

- * Correlation analysis was performed to identify relationships between features.

- * Predictive power matrix was calculated to assess the relevance of each feature for predicting wind speed.

iii. Feature Engineering

- * Features and target variable were separated.

- * Data was split into training and testing sets.

- * A pipeline was created to standardize the data, select relevant features using mutual information, and reduce dimensionality using PCA.

iv. Model Training and Evaluation

- * Three models were trained on the transformed training data: Random Forest, XGBoost, and a Neural Network.

- * These models were then combined into an ensemble model using a voting regressor.

- * The ensemble model was evaluated on both the training and testing sets using various performance metrics such as mean squared error, mean absolute error, R2 score, and root mean squared error.

v. Model Persistence


The trained pipeline and ensemble model were saved to disk using joblib for future use.

c. Additional Notes

- * The section code includes code for making predictions on new observations.

- * The documentation provides a clear and concise summary of the steps involved in the analysis and modeling process.

5. App deployment

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	12 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

The application features a user-friendly interface, allowing users to input hurricane data in real-time. Upon calculation, the estimated wind speed is displayed alongside the hurricane's global location on a map.

6. Cloudera's AMP Platform Structure

This real-time functionality alongside Cloudera's solid infrastructure and Machine Learning empowerment hold promise for proactive decision-making, enabling timely responses to escalating hurricane categories based on updated data.

6. Future Work

The following are some areas for future work:

1. Fine-tune the model hyperparameters.
2. Collect additional data to improve the model's accuracy.
3. Apply different time series models, this is linked to the polishing of the original dataset where time data can have no irregularities that may not affect the accuracy of the model with respect to the actual observed values.
4. Improve the user interface by allowing multiple values to be estimated at different points in time and linking it to a streaming database.

7. References

- <https://www.nhc.noaa.gov/data/tcr/index.php>


- <https://www.wunderground.com/article/news/news/2023-07-14-noaa-hurricane-forecast-model>

- <https://www.noaa.gov/news-release/noaa-launches-new-hurricane-forecast-model-as-atlantic-season-starts-strong>

- <https://hfip.org>

- <https://www.nhc.noaa.gov/aboutsshws.php>

- <https://www.visualcrossing.com/weather-history/11.7,157.1/metric/2018-09-29>

	Code of Duty Department Revision Division	SOP #	1
		Revision #	1
		Implementation Date	06/03/2024
Page #	13 of 13	Last Reviewed/Update Date	06/03/2024
SOP Owner	Code of Duty	Approval	Approved

-<https://dev.meteostat.net/python/hourly.html#api>

-<https://hurricanescience.org/science/forecast/models/modelskill/index.html>

-<https://yaleclimateconnections.org/2023/06/which-hurricane-models-should-you-trust-in-2023/>

-<https://www.wfla.com/weather/tracking-the-tropics/tracking-the-tropics-how-accurate-are-hurricane-season-forecasts/>

-<https://web.mit.edu/12.000/www/m2010/teams/neworleans1/predicting%20hurricanes.htm>

-<https://yaleclimateconnections.org/2023/10/nightmare-scenario-category-5-hurricane-otis-devastates-acapulco/>

8. Contributors

Aarón Castillo Medina - Data Engineer and ML Engineer

- ❖ aaroncastillo329@gmail.com
- ❖ <https://www.linkedin.com/in/amcm329/>

Javier Sánchez Silva - Data Scientist

- ❖ javiermega2@gmail.com
- ❖ <https://www.linkedin.com/in/javier-amiel-irais-sánchez-silva-86a309167>

Gilberto Subias García - Data Scientist

- ❖ subias.gilberto@gmail.com
- ❖ <https://www.linkedin.com/in/gilberto-s-a64757121/>