REPORT

Twins of The Winds

(Summer of Innovation 2024)

TEAM: CLOWNS 🤡

Aniruddh Pandav (Team Lead)

Maitreyee Kumbhojkar

Vidit Parikh

Contents:

Approach:

Contains a brief description of the thought process and approach towards the problem

Detailed analysis of each file:

Approach:

Given the datasets and features for the years 1980-1996, we first sorted the data chronologically. This facilitated trend analysis and the application of time series models and also managed the larger data volumes in the later years compared to the 1980s.

Next, we focused on selecting the best NaN value handling methods for time series and seasonality-dependent data. We used spline interpolation for a few columns and XGBoost for 'humidity'.

For column transformations, we applied Yeo-Johnson and log transformations to approximate a Gaussian distribution. This aids in handling test datasets with similar KDE but different values.

Feature selection could have been improved with data on El Niño/La Niña occurrences, as these events directly impact sea surface temperature. Attempts to find consistent data online were unsuccessful. The 'day' feature was dropped since sea surface temperature depends more on the month and year.

Model training and testing began with XGBoost, ARIMA, and LSTM. XGBoost initially outperformed the others, but later, the Random Forest Regressor performed better on an 80% train/test split. However, increasing the training set size significantly improved XGBoost's performance on test and cross-validation sets, whereas Random Forest was unaffected.

After evaluating the bias and variance of the models, we chose XGBoost with tuned hyperparameters (using Grid Search CV) and an optimized training set size as the final model. Its parameters were saved in pickle format.

We used this pickle file to predict sea surface temperature for the test datasets, and the results were saved in CSV format. The data is in chronological order, so please follow the instructions in Readme.md for proper evaluation.

Analysis of Files:

analysis_training_testing/Analysis.ipynb:

The following ipynb file investigates importing the CSV files into the notebook. The tasks further performed after observing this data are – identifying the dimensions of the input, identifying if there are null values in a particular column (if yes, how many?), representing the data in a graphical format and trying to draw a hypothesis about what model could work for that kind of data, etc.

analysis_training_testing/lstm.ipynb:

This file has the results when we ran the Long Short-Term Memory (LSTM) Model on the provided data post the pre-processing of the same. Some of the highlights were:

- Total parameters (all of them were utilised for training): 74651
- Number of epoch: 30
- Output: 51 parameters dense
- Test loss: 5.579498767852783
- Test RMSE: 2.3631013149253746
- R2 score: -0.05332094734542392

It is quite evident from the R2 values that the LSTM model for this kind of data is not suitable which inspired us to work further.

analysis_training_testing/model_selection.ipynb:

The provided Jupyter notebook explores a dataset involving environmental variables like wind speeds, temperature, and humidity, aiming to predict sea surface temperatures. After importing and exploring the dataset, key tasks included handling null values, interpolating missing data, and visualizing temporal trends and distributions of features. Preprocessing involved scaling and transforming features for machine learning models, specifically XGBoost and Random Forest regressors, optimized via GridSearchCV. The models were evaluated using metrics like MSE and R^2 score (0.915), with considerations for overfitting. Insights included the impact of temporal data trends on model performance and strategies for handling missing data in predictive modelling.

analysis_training_testing/model_selection_2.ipynb:

To address the task of predicting temperature changes, we employed advanced preprocessing and modeling techniques on a dataset sourced from climate monitoring. Initially, data was split sequentially into training and test sets based on time elapsed, ensuring that the model is trained on earlier data and tested on later observations. Various transformations such as Yeo-Johnson power transformation for wind zones and standardization for other features like air temperature were applied to normalize the data. Hyperparameter tuning was conducted using XGBoost regression, optimizing parameters like learning rate, maximum depth, and number of estimators through GridSearchCV. The best model configuration achieved a mean squared error of 0.21, indicating strong predictive performance. Evaluation metrics such as R-squared further confirmed the model's robustness, achieving an impressive score of 0.96 on the test data. This model, finely tuned and validated, was then serialized using pickle for deployment and future use.

analysis_training_testing/plan.txt:

This document consists of our initial idea after a preliminary round of investigation of the data and brief research on the provided topic. The methodology was further refined as we moved on with the statistical results and adapted accordingly to our obtained values.

analysis_training_testing/testdata_analysis.ipynb:

In this analysis, we compared three datasets: 'data_1997_1998.csv', 'evaluation.csv', and 'updated.csv', focusing on key meteorological features. After preprocessing, which included dropping unnecessary columns, we visualized the distributions and outliers for features such as year, month, day, latitude, longitude, zonal winds, meridional winds, humidity, and air temperature. Using histograms with kernel density estimation, we compared the distributions across the datasets, revealing notable differences and similarities. Additionally, box plots highlighted the presence of outliers, crucial for understanding data quality and variability. The visualizations aimed to guide the decision on applying transformations like log, box-cox, or yeo-johnson on features such as zonal winds and air temperature for improved model performance. This comprehensive analysis is a step toward refining our predictive models by understanding and addressing data inconsistencies and outliers.

datasets/train.csv

This is the file provided as the primary dataset for this competition.

datasets/updated.csv

This dataset has modifications after preprocessing data and can be used directly for training

1997_1998_predictions.ipynb

In this data preprocessing and modeling task, we began by loading and sorting weather data spanning 1997-1998. Initial data cleaning involved converting columns to appropriate data types and sorting by date. Missing values were interpolated for key weather features using spline interpolation. Subsequently, the dataset was split into known and unknown humidity values. A grid search with cross-validation was employed to tune an XGBoost regressor to impute missing humidity values in the dataset. After ensuring no missing values remained, transformations were applied to normalize the data, specifically using the Yeo-Johnson method for 'zon.winds' and 'air temp.', followed by standard scaling. The processed data was then used with the trained XGBoost model to predict sea surface temperatures, and these predictions were saved to a CSV file for further analysis. This comprehensive approach ensured a robust and accurate prediction model while maintaining data integrity and consistency.

evaluation_prediction.ipynb

Firstly, the dataset evaluation.csv is loaded into a Pandas DataFrame. After sorting the data chronologically and dropping unnecessary columns (date, Unnamed: 0, and day), missing values in the features air temp., zon.winds, mer.winds, and humidity are handled. Specifically, interpolation using spline method of order 3 is applied to air temp., zon.winds, and mer.winds to fill missing values.

Next, the dataset is split into two parts: one with known values (df_known) and one with missing humidity values (df_nan). A machine learning model, XGBoost regressor, is trained on df_known using grid search with cross-validation to predict humidity based on features like month, year, latitude, longitude, and air temp.. After identifying the best model parameters through grid search, predictions are made for the missing humidity values in df_nan using this model.

After imputing missing values, the dataset undergoes transformation steps. Features zon.winds and air temp. are transformed using the Yeo-Johnson power transformation and then standardized using StandardScaler.

Finally, the trained XGBoost model (xgb_best_model.pkl) is loaded from disk, and predictions are generated for sea surface temperatures (s.s.temp) based on the transformed dataset. These predictions are saved to a CSV file named prediction_evaluation.csv.

Overall, this workflow involves data preprocessing, missing value imputation, model training using XGBoost with hyperparameter tuning, data transformation, and prediction generation, showcasing a comprehensive data science pipeline.

final_train.ipynb

The provided code performs data preprocessing and model training for a machine learning project. It starts by importing necessary libraries such as pandas, seaborn, matplotlib, and numpy for data manipulation and visualization. The dataset is read from a CSV file and sorted in chronological order based on a newly created date column. Initial handling of missing values is done using spline interpolation for certain features, and the XGBoost algorithm is employed to impute remaining missing values in the humidity column. The data is then saved to an updated CSV file.

Subsequently, further preprocessing is applied, including feature selection and transformations. The 'time_elapsed' and 'day' columns are dropped, and power transformations are applied to 'zon.winds' and 'air temp.' features using the Yeo-Johnson method. The dataset is scaled using StandardScaler. The model training phase uses the best parameters obtained from a previous grid search to initialize an XGBoost classifier. The model is trained on the preprocessed dataset and saved using pickle for future use.

The provided code demonstrates a comprehensive workflow for data preparation, handling missing values, and model training with XGBoost, ensuring that the final model is ready for deployment.

prediction_1997_1998.csv

This is the predicted outcomes file of our model for the data provided for the years – 1997-98.

The **predicted values** are **sorted** in **chronological order**.

prediction_evaluation.csv

This is the predicted outcomes file of our model for the years 1980-1996.

The **predicted values** are **sorted** in **chronological order**.

xgb_best_model.pkl

As you could read from above that we found XGB model to be performing the best for the given data, we have considered saving it in a pickle file and exported it here for further perusal.

Thank You!