TWINS OF THE WINDS – TEAMNAME

AIM

To develop a machine learning model on labelled data that can take sequential data and generate Sea Surface Temperature and then predict the same for the unlabelled data.

DATA PROVIDED

1. **train.csv** – contains the labelled data along with the target variable (sea surface temperature)
2. **evaluation.csv** – contains unlabelled data for the years 1980-1996
3. **data_1997_1998.csv** – contains unlabelled data for the years 1997 and 1998

NATURE OF DATA PROVIDED

1. **Date**: Day, Month, Year when the observation taken place from the Buoy.
2. **Latitude and Longitude**: Location of Buoy during the observation.
3. **Wind Data**: Two Wind data we have Zonal Wind and Meridional Wind.
4. **Humidity**: Relative Humidity during observation

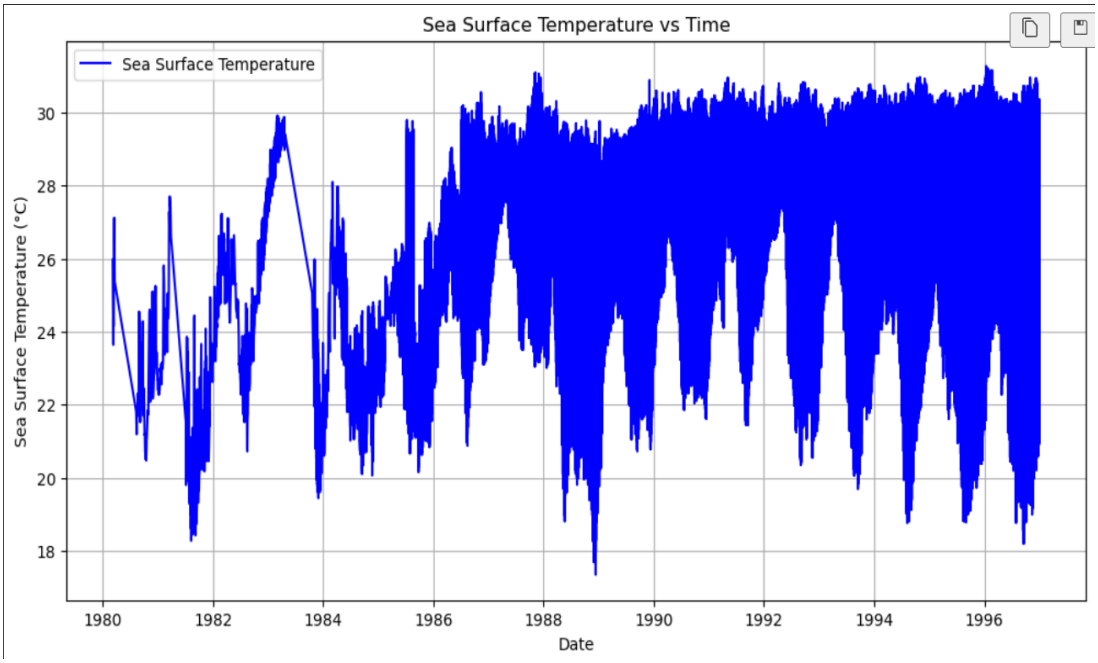
5. **Air Temperature**
6. **Sea Surface Temperature** (Target Variable)

PREPROCESSING

- Imported some basic libraries like pandas, numpy, matplotlib.pyplot and sklearn.
- The train.csv file was read and stored in a dataframe.
- A correlation matrix was plotted to check correlation amongst the factors and also with that of the sea surface temperature (s.s.temp).
    1. Best correlation of sea surface temperature was with air temperature – 0.95
    2. Moderate correlation was observed with longitude, zonal winds , meridional winds and humidity.
    3. Low correlation was with day, month, year and latitude.
- Dataset was checked for the presence of null values and it was found that zonal winds and meridional winds both had 14570 null values , humidity had 38794 null values and air temperature had 7838 null values.
- The data we received was in a randomised order. We concluded that since we had to deal with null values (which by any method used, would be based on the values of the data present in its surroundings) and also understand the nature of the features, we had to rearrange the data according to date and time. To do so, we converted the day, month and year columns into a

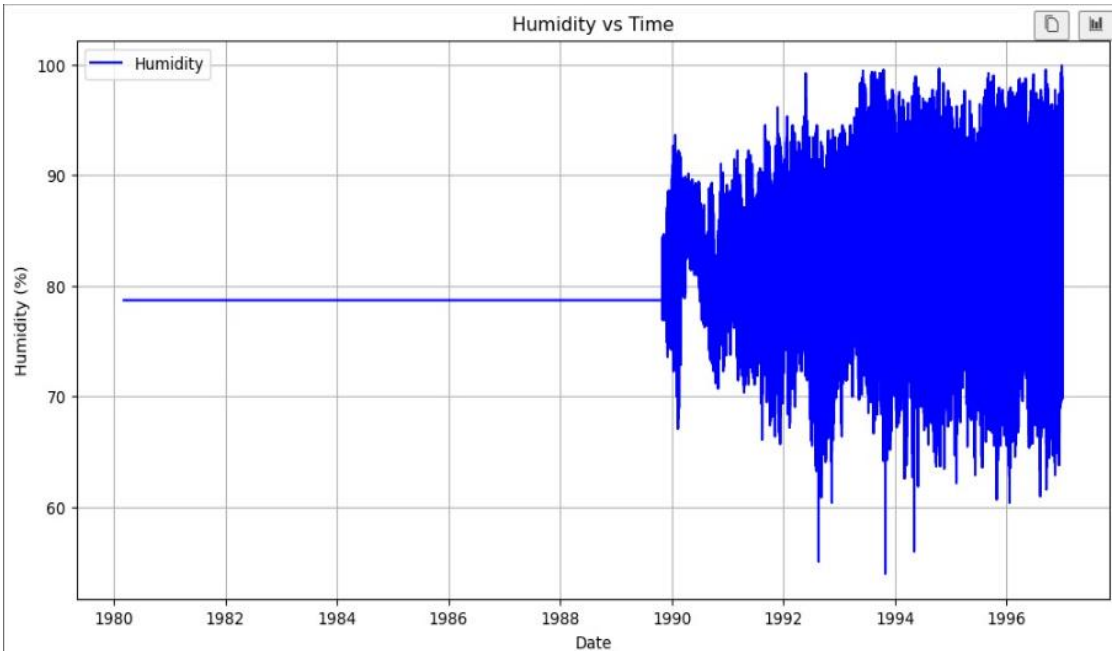
single one and called it the ‘date’ column. This proved to be an easier and less tedious method.

- Once we were done with the data organisation, we plotted graphs of each of the features against time to observe their variation. Sea surface temperature showed a clear variation with time and hence we decided to include the time features – day, month and year in our model even though they showed lesser correlation.

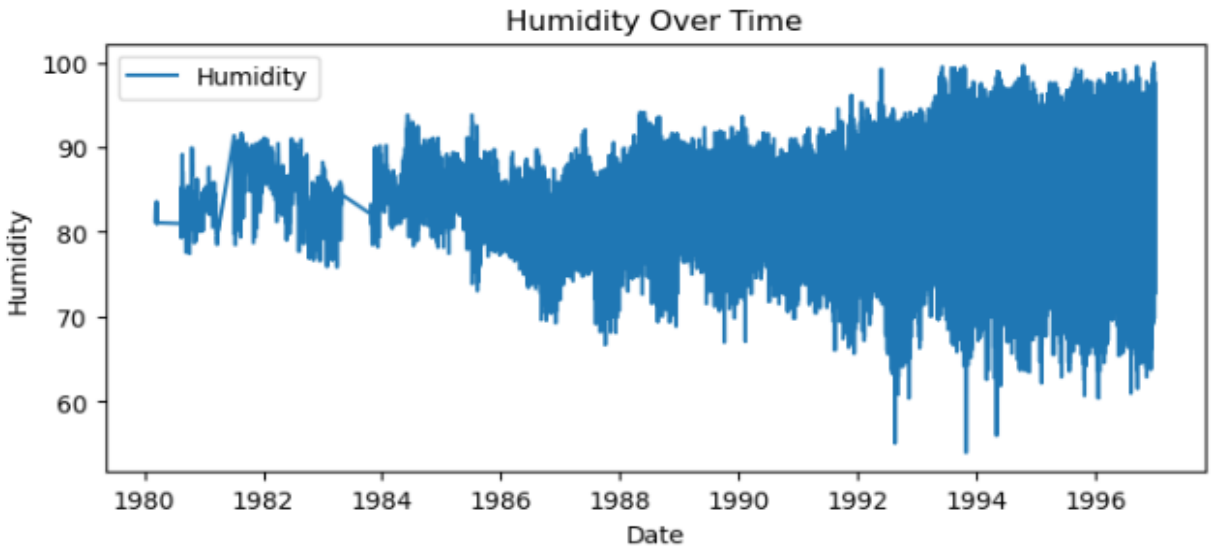


- Since the number of missing values was too high, dropping the rows having them was not ideal. Hence the first method we approached was interpolation for humidity and the use of mean for the other features. The column humidity had about first 15000 odd rows as null which were not filled by interpolation and to deal with this, we used bfill (backward filling).
- But when the graphs of the features against time were plotted again, the one for humidity had an initial

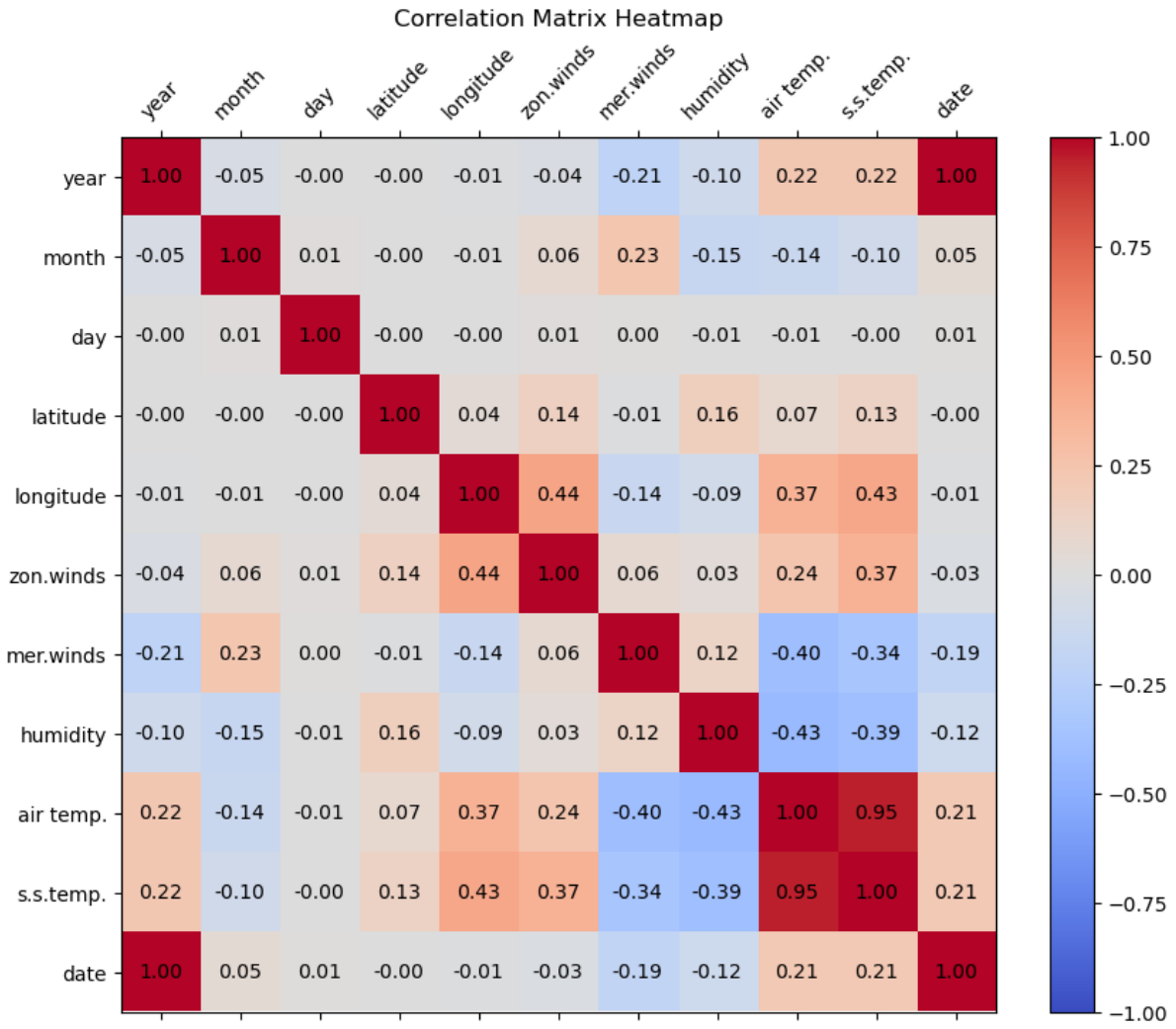
constant line which represented the initial rows having the same value filled by bfill.



- While checking for more efficient ways of filling in the missing values, we encountered the **KNN Imputer**. The KNN Imputation technique is used to fill in missing values by considering a given K nearest neighbours of the data point with the missing value. The number of neighbours we used here was 5 as the error decreases when k value increases until a point after which error rate goes on increasing with increasing k value. Hence, we made our choice of k value as 5.
- The graphs for the features were plotted again and this time the humidity graph showed a considerable variation which was more realistic than the previous one. Hence, we decided to stick with the KNN Imputer for filling our null values.



- We plotted another correlation matrix and made our analysis. Now that we had dealt with null values, the correlation of sea surface temperature with humidity had increased (although by a small margin). There were small increments in correlation factors of other features too.



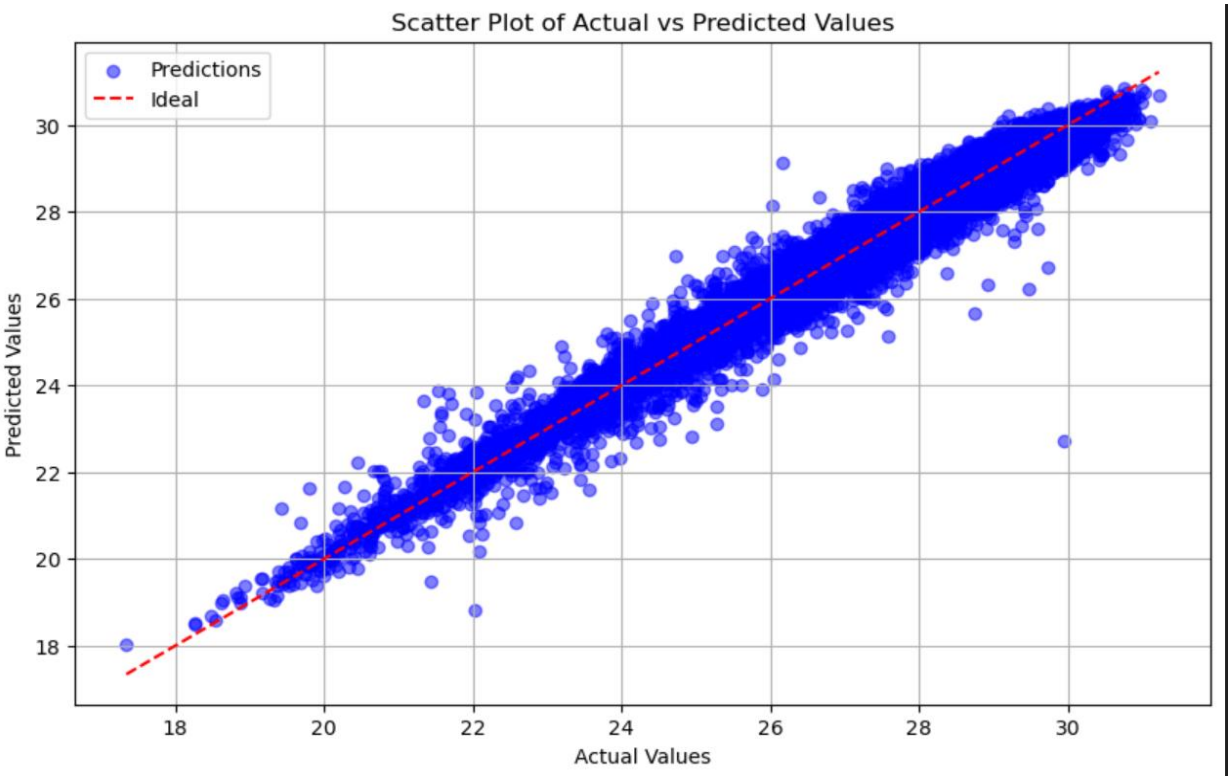
- Once we realised we had a positive increment, we concluded our data to be sufficiently preprocessed to be put into our model.

MODEL SELECTION AND TRAINING

- We used the **Random Forest Regressor** model to train our data. We chose this model due to multiple reasons. Firstly, it is an ensemble machine learning method that is based on decision trees and combines multiple of them to make more accurate predictions required during

time series forecasting (which is the nature and requirement of our data too). It is a powerful technique that predicts future values based on historical data. Secondly, it requires much less input preparation. This means that it can handle categorical as well as numerical features without any feature normalization.
Lastly, this model is much quicker to train and to optimize according to their parameters.

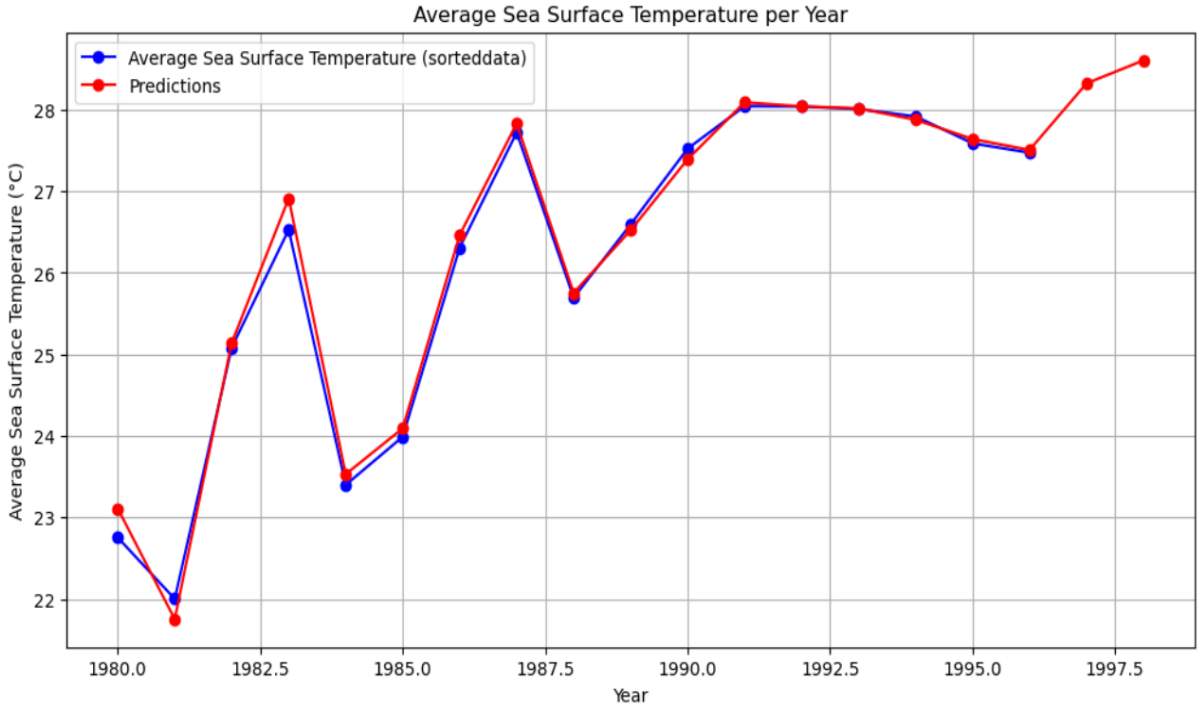
- For the training process we selected day, month, year, latitude, longitude, zonal winds, meridional winds, humidity and air temperature as our features.
- The target variable was of course sea surface temperature.
- We split our data into two parts, 'train' and 'test' with 'test' having 20% of the data. This was done to train the model on the 'train' part and check its working and make predictions on the 'test' part.
- The predictions made were obtained in the form of an array and the accuracy of the model was checked using the factors, Mean Absolute Error, Mean Squared Error and the R2 Score.
- **Mean Absolute Error – 0.1153310897064457**
  Mean Squared Error – 0.1153310897064457
  R2 Score – 0.975490442231400
- The scatter plot for the actual vs predicted values was plotted and it looked as such.



EVALUATION OF UNLABELLED DATA (TEST DATASET)

- The processes are similar to those followed in the train dataset.
- We have two csv files – evaluation.csv and data_1997_1998.csv . The data is read from these files and stored in two separate dataframes.
- The first column of each of the csv files were dropped, one being ‘Index’ and the other unnamed.
- The graphs for features against time were plotted for each of the dataframes. Similar variations were observed in the two of them. Hence, we merged the two dataframes into a single one. This made dealing with the data easier.

- This data also was first converted into date format and arranged sequentially.
- The null values were checked and were found to be present in the columns zonal winds, meridional winds, humidity and air temperature. They were filled in by the usage of KNN Imputer as done previously.
- While feeding in this data in the model, we used the same features as before.
- Predictions were made and stored in the form of an array.
- We sorted them in a file and to check how accurate our predictions were, we plotted the trend of the average sea surface temperatures (per year) predicted by our model along with that of the actual ones present in the train dataset against the years.
- From the graph we concluded that the predicted temperatures were quite similar to the actual temperatures and hence our model working was quite accurate.
- We also could predict and plot the average sea surface temperatures for the years 1997 and 1998.


Average Sea Surface Temperature per Year
Average Sea Surface Temperature (sorteddata)
Predictions
Average Sea Surface Temperature (°C)
Year
22
23
24
25
26
27
28
1980.0
1982.5
1985.0
1987.5
1990.0
1992.5
1995.0
1997.5