

**SEA SURFACE TEMPERATURE
ANALYSIS AND PREDICTION IN COSTAL
TAMILNADU**

Submitted in partial fulfillment of the requirements for
the award of

Bachelor of Engineering degree in Computer Science and Engineering
by

ANNAPUREDDY MALLAREDDY (Reg. No. 37110042)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

**Accredited with Grade “A” by NAAC JEPPIAAR NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600 119**

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **ANNAPUREDDY MALLAREDDY(Reg. No. 37110042)** who carried out the project entitled **“SEA SURFACE TEMPERATURE ANALYSIS AND PREDICTION IN COSTAL TAMILNADU”** under my supervision from **November 2020** to **March 2021**.

Internal Guide

Dr. P.SARDAR MARAN, M.E.,Ph.D

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **AVUTHU HEMANTHREDDY** hereby declare that the Project Report entitled “**SEA SURFACE TEMPERATURE ANALYSIS AND PREDICTION IN COSTAL TAMILNADU**” is done by me under the guidance of **Dr.P.SARDAR MARAN, M.E.,Ph.D** Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing,

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D .**, Dean, School of Computing and **Dr.L.Lakshmanan M.E., Ph.D.,and Dr.S.Vigneshwari M.E., Ph.D.**, Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.P.SARDAR MARAN M.E.,Ph.D Associate Professor**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Increase in mean sea surface temperature (SST) over the years is the primary indicator of ocean warming; which is likely to impact fish growth, reproduction and migration in the ocean. As a prelude to understanding changes in biotic communities along the Tamil Nadu coast of India, the trend in SST along the Tamil Nadu coast was assessed over a period of 105 years (1906-2010). The annual catch data of two major pelagic resources, the Indian oilsardine and Indian mackerel, and the annual catch (1979-2010) by the trawlers operating from Madras Fisheries Harbour, Chennai, were correlated with SST. There was an increase in minimum and maximum SST in all the seasons off Nagapattinam and Kanyakumari while at Chennai, the minimum SST in south-west monsoon season and maximum SST in summer season has decreased. The catches of oilsardine and mackerel showed an increasing trend over the last 25 years. Seasonal analysis showed positive correlation of the catches with SST. The catch of oilsardine during the summer months in particular showed an increasing trend over the period. Among the 17 dominant resource groups assessed, the catch of whitebaits showed significant positive correlation with increasing SST (at 5% level). Ribbonfish, perches and crabs also showed high 'r' values but the correlation was not statistically significant.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	viii

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 WHAT IS SEA SURFACE TEMPERATURE	2
	1.2 OVERVIEW	4
	1.3 APPLICATIONS	5
2.	LITERATURE SURVEY	6
	2.1 LITERATURE REVIEW	6
3.	AIM AND SCOPE	10
	3.1 AIM OF THE PROJECT	10
	3.1.1 OBJECTIVES	10
	3.2 SCOPE	11
	3.2.1 ADVANTAGES	11
	3.3 SEA SURFACE TEMPERATURE ANALYSIS	11
4.	SOFTWARE DEVELOPMENT METHODOLOGY	13
	4.1 DESCRIPTION OF DIAGRAM	13
	4.2 SOFTWARE ENVIRONMENT	16
	4.3 ALGORITHM USED	20
	4.4 MODULES USED	30
	4.5 ARCHITECTURAL DESIGN	34

5.	RESULTS AND DISCUSSION	35
6.	CONCLUSION AND FUTURE WORK	39
	REFERENCES	39
	APPENDIX	
	A. SOURCE CODE	41
	B. SCREEN SHOT	43
	C. PALAGRISM REPORT	48

LIST OF FIGURES

Figure-number	name	Page number
1.1.1	Sea surface temperature	3
1.1.2	Sea surface temperature graph	3
4.1.1	System design	13
4.3.1	Our labeled data	21
4.3.2	Hyper plane is simply a line	22
4.3.3	A more complex data set	23
4.3.4	Before division	24
4.3.5	After division	24
4.3.6	Everything is neatly separated	25
4.3.7	Low regularization value	27
4.3.8	High regularization value	27
4.3.9	High gamma	28
4.3.10	Low gamma	28

4.5	ARCHITECTURAL DESIGN	34
5.1	Data set	36
5.2	Trained data set	37
5.3	Output	37
5.4	Accuracy	38
6.1	Importing libraries	43
6.2	Performing operation	44
6.3	Taking variables	44
6.4	Importing algorithms	45
6.5	Predicting x variable	45
6.6	Result	46

CHAPTER 1

INTRODUCTION

INTRODUCTION

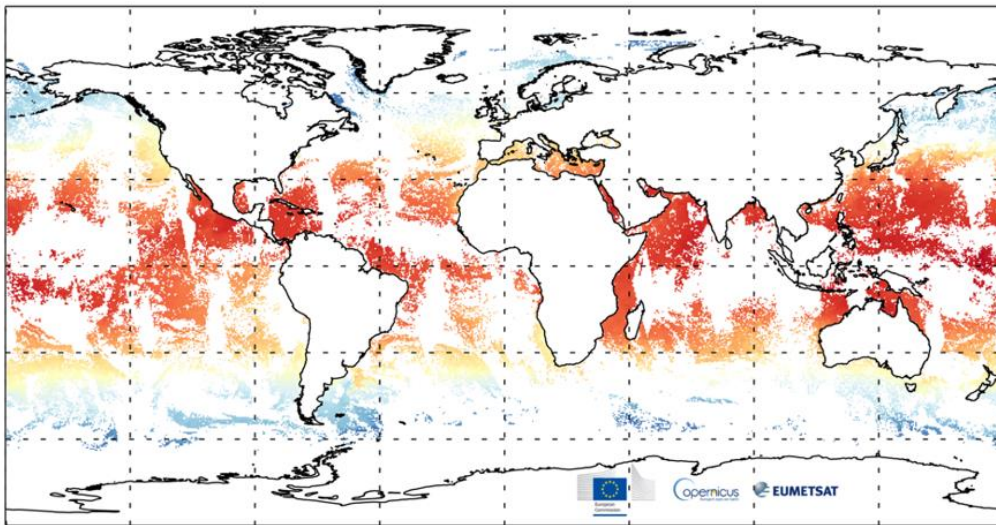
Increase in mean sea surface temperature (SST) over the years is one of the primary indicators of global warming. Climate change is undoubtedly complicating the patterns in global fisheries. The evidence of global warming is a 0.6°C increase in mean annual SST over the past 100 years (Karl and Trenberth, 2003). Temperature affects other ocean parameters such as salinity, pH and dissolved oxygen. All these factors have a synergistic effect on the biota ranging from microscopic plankton to large fishes and marine mammals. Physiological functions like growth, reproduction, spawning, and recruitment are directly influenced by change of temperature (Chowdhury et al., 2010). Primary production, secondary production, fish distribution (through permanent movement or changes in migration patterns), fish abundance (primarily due to changes in primary and secondary production) and phenology, i.e., timing of life-cycle events such as spawning are also affected by change of temperature (Harvell et al., 1999; Beaugrand et al., 2002; Edwards and Richardson, 2004; Perry et al., 2005; Edwards et al., 2006; Lehodey et al., 2006; Bruno et al., 2007; Reid et al., 2007). Studies from the Indian sub-continent on the impact of changes in SST on fish production are minimal and mainly focused on extension of distribution range of the oilsardine (Vivekanandan et al., 2009; Vivekanandan and Krishnakumar, 2010) and Indian mackerel (Asokan et al., 2009), shift in spawning peaks in the threadfin bream, *Nemipterus japonicus* (Vivekanandan and Rajagopalan, 2009), coral bleaching (Vivekanandan and Jeyabaskaran, 2010) and vulnerability of marine ecosystems and fisheries (Vivekanandan, 2009; 2010; 2011). In the present study, as a prelude to understanding changes in biotic communities along the Tamil Nadu coast of India induced by alterations in climatic conditions, annual catch data of major fishery resources were correlated with SST. Data on SST (9 km resolution) were downloaded from the website of International Comprehensive Ocean-

Atmosphere Data Set (ICOADS), NOAA/NASA (BSRL PSD www.cdc.noaa.gov) following standard protocol. Three regions, Chennai, Nagapattinam and Kanyakumari, were selected for retrieving the SST data over the last 105 years (1906-2010). Months were grouped into four seasons, based on the climate conditions in Tamil Nadu: post-monsoon (January-March), summer (April-June), south-west (SW) (July-September) and north-east (NE) monsoon (October-December). The data were sorted season-wise for four seasons: and mean value and anomaly for each season of every 20 years were calculated for all the three regions. Pooled data were used to arrive at a profile of the SST along Tamil Nadu coast. Available fish catch data with the Fisheries Resources Assessment Division of the Central Marine Fisheries Research Institute (CMFRI), Kochi were tabulated along with the SST and correlations were made. Annual catch data (all gears, for the period 1985-2010) of two major pelagic resources, Indian oilsardine and Indian mackerel, were correlated (Pearson correlation) with SST. Trawl catch from north Tamil Nadu coast by trawlers operating from Madras Fisheries Harbour, Chennai, was also related to the average annual SST over a period of 32 years (1979-2010). The analysis was done for the total trawl catch as a whole and for the catch of oilsardine, Indian mackerel, threadfin breams, lizardfishes and croakers, and the Pearson correlation coefficient 'r' was estimated. The exercise was repeated with a recent set of data (1998-2010) for total catch of major fishery resources landed by all gears and the average SST along Tamil Nadu coast.

1.1 Sea surface temperature (SST)

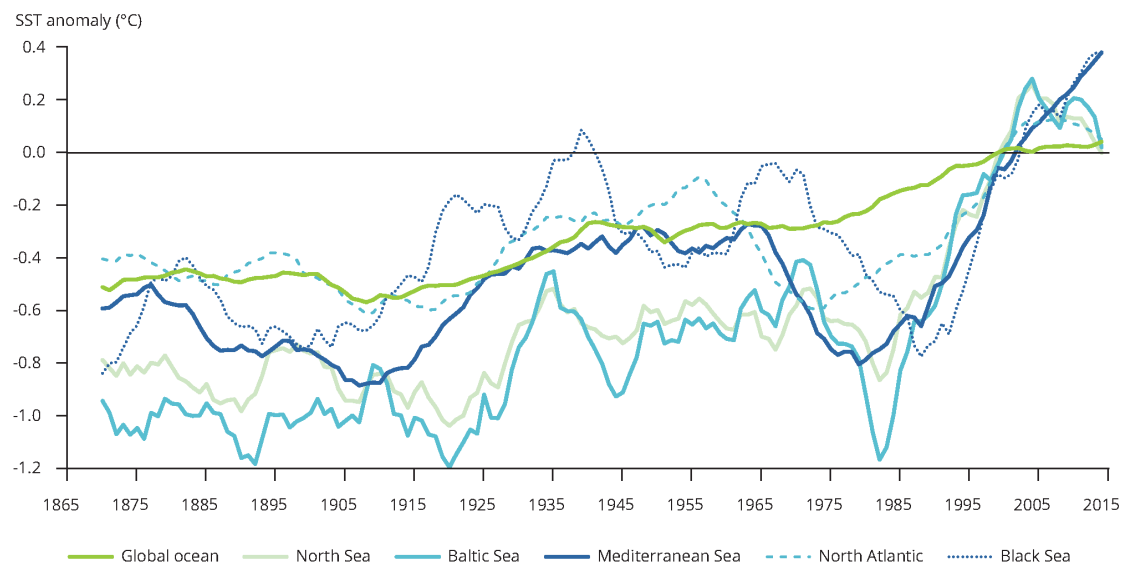
Sea surface temperature (SST) is the water temperature close to the ocean's surface. The exact meaning of *surface* varies according to the measurement method used, but it is between 1 millimetre (0.04 in) and 20 metres (70 ft) below the sea surface. Air masses in the Earth's atmosphere are highly modified by sea surface temperatures within a short distance of the shore. Localized areas of heavy snow can form in bands downwind of warm water bodies within an otherwise cold air mass. Warm sea surface temperatures are known to be a cause of tropical cyclogenesis over the Earth's oceans. Tropical cyclones can also cause a cool wake, due to turbulent mixing of the

upper 30 metres (100 ft) of the ocean.



1.1.1 Sea surface temperature

Sea surface temperature (SST) changes diurnally, like the air above it, but to a lesser degree. There is less SST variation on breezy days than on calm days. In addition, ocean currents such as the Atlantic Multidecadal Oscillation (AMO), can effect SST's on multi-decadal time scales,[4] a major impact results from the global thermohaline circulation, which affects average SST significantly throughout most of the world's oceans.



1.1.2 Sea surface temperature graph

1.2 OVERVIEW

Sea surface temperature (SST) is a strong indicator of productivity, pollution, and global climate change, and this can be measured using thermal infrared (IR) bands from optical satellites. Interpretive techniques are required to convert the data to actual temperature. Sea surface temperature SST has been used to identify sources of pollution, thermal pollution, upwellings, and areas of variant productivity. SST is also an indicator of stress to corals and other species.

Sea surface temperature (SST) is an essential climate variable for understanding the climate system and quantifying ongoing climatic change. Global mean SST has risen from decade-to-decade since the 1970s, with consequences for global weather patterns and oceanic ecosystems. Most striking is the increasing frequency of mass bleaching events of coral reefs. Long-term climate data records of SST have been developed from satellite data that confirm the quantification of climate change seen in in situ measurements.

Sea surface temperature (SST) refers to the state of the uppermost 'mixed' layer of the ocean, typically only few tens of meters thick. Knowledge of past SST variability is important as it helps us understand how the ocean behaves in times of climate change; it allows us to validate numerical climate models and assess the significance of current climate trends. Temperature affects a range of thermodynamic, metabolic, and biological processes that leave their signature in the geological record. Past SST can be reconstructed from fossil remains of phyto- and zooplankton in marine sediments and from skeletons of hermatypic corals. Geochemical paleothermometers are based on the analysis of oxygen isotopes and Mg/Ca in planktonic foraminifera and oxygen isotopes and Sr/Ca in corals. Biomarker methods are based on the measurement of various types of alkenones produced by haptophyte algae and crenarchaeotal membrane lipids in the organic matter in sediments. The composition of microfossil assemblages (planktonic foraminifera, diatoms, radiolarians, and dinoflagellate cysts) in modern sediments is used to develop transfer functions to convert census counts in fossil samples to SST. The standard errors of environmental calibrations of paleothermometers range between 0.5

and 2 °C; resolution of SST reconstructions in marine sediments is typically 10–100 years, while corals allow yearly to monthly resolution

1.3 APPLICATIONS

Because the ocean covers 71 percent of Earth's surface, scientists record sea surface temperature (SST) to understand how the ocean communicates with Earth's atmosphere. SST provides fundamental information on the global climate system. SST is an essential parameter in weather prediction and atmospheric model simulations, and is also important for the study of marine ecosystems.

SST data are especially useful for identifying the onset of El Niño and La Niña cycles. During El Niño, temperatures in the Pacific near the equator are warmer than normal. During La Niña, the same area experiences colder than normal ocean temperatures. These cycles are caused by multiyear shifts in pressure and wind speeds, and affect ocean circulation, global weather patterns, and marine ecosystems.

SST measurements benefit a wide spectrum of operational applications, including climate and seasonal monitoring/forecasting, military defense operations, validation of atmospheric models, sea turtle tracking, evaluation of coral bleaching, tourism, and commercial fisheries management.

CHAPTER 2

LITERATURE SURVEY

LITERATURE REVIEW

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

2.1 A novel solution for high speed internet over the oceansThe cellular network range over the oceans is limited to about 15 km from the shore in most places. The service providers do not have any incentive to extend the coverage further. However, marine fishermen who routinely spend 5 to 7 days at the ocean on a single fishing trip are severely impacted by this. They go as far away as 120 km from the shore on some occasions and are completely cut off from the land during their fishing trips. Also, they are generally poor and need an economical solution to stay connected. We have developed and successfully prototyped an affordable solution based on a multilevel auto-reconfigurable backhaul infrastructure network. It uses long range (LR) Wi-Fi and extends the connectivity to 60 km and beyond over the oceans using heterogeneous networks and relay nodes. Multiple field trials involving up to four boats over the Arabian Sea have yielded consistent and repeatable results even under rough sea states. Additional experiments conducted over the backwaters by scaling up the network further have also yielded positive results. This work provides a detailed description of the architected solution and the results obtained.

2.2 PREDICTION OF SEA SURFACE TEMPERATURE IN THE SOUTH CHINA SEA BY ARTIFICIAL NEURAL NETWORKS

Sea surface temperature (SST) significantly affects the processes of air-sea interactions and thus forms an important indicator of climate changes. In SST predictions, the approach of artificial neural networks (ANNs) is data-driven, unlike that of the numerical models, which are physics-based. In this paper, Operational Sea Surface Temperature and Ice Analysis (OSTIA) dataset was used for training ANN models and verifying prediction results. To reduce the prediction error caused by SST variations, the authors

propose to separate SST time series data into climatological monthly mean and monthly anomaly datasets, and construct two neural network models. The combination of these two models gives the final SST prediction results. This method was used to predict the SST in the South China Sea. The average bias and standard deviation between the predicted SST and OSTIA SST are $-0.02\text{ }^{\circ}\text{C}$ and $0.37\text{ }^{\circ}\text{C}$, respectively. The results indicate that the proposed training method gives good prediction accuracy.

2.3 Basin-Scale Prediction of Sea Surface Temperature with Artificial Neural Networks

Prediction of sea surface temperature (SST) is desired for several applications ranging from climate studies to maintenance of coastal ecosystem. Such prediction with the help of artificial, or simply, neural network has by now fairly stabilized. However corresponding studies are mostly applicable only to a specified single location. In this study we have expanded them to cover an entire sea basin. The basin under consideration is Bay of Bengal (BoB) located on the east side of the Indian peninsula. We have predicted SST at the daily time scale using time series approach in which we feed a selected length of past daily SST observations to the neural network and derive the predicted value of SST at multiple lead times (days) as output. The gridded NOAA v2 high resolution dataset derived from satellites was used for this purpose. At every grid in the BoB feed forward back propagation type of neural network was developed. The networks were trained using 70 % of data and tested with the help of remaining 30 %. The performance in testing of such large spatial-scale networks was judged on the basis of the error statistics of correlation coefficient, 'r', and root mean square error, RMSE. The prediction skill of ANN models were found to be very good at shorter lead times (1-3 days)

and reasonably good at higher lead times (4-7 days). Apart from that, these ANN models were also evaluated for their performance during extreme weather events which are peculiar to BoB region and found to be capturing such events in advance with sufficient time. Overall therefore it is claimed that the basin-scale neural networks developed in this study can not only carry out multiple time step predictions of daily SST at individual grid points simultaneously but can also predict basin scale weather phenomena in advance.

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF THE PROJECT:

The main aim of the project is to predict the temperature of a Sea on a specific date. Then train the data using machine learning algorithms and split them in the ratio 70:30 by using SVM algorithm. Where the 70% is trained and remaining 30% is tested based on the trained data. Then by using all the data we can predict the data for upcoming days.

3.1.1 OBJECTIVE:

Sea surface temperature (SST) is and has been one of the most measured variables in the ocean, and as such it has received much scientific attention. Using a combination of ship SSTs, moored and drifting buoy SSTs, and satellite infrared SSTs, we have characterized the global and seasonal patterns of SST. Realizing that only the satellite infrared SSTs can represent the SST of the 10 μm thick skin layer of the ocean, future efforts should be aimed at separating the satellite skin SSTs from the 1–5 m deep bulk SSTs measured by the buoys and ships. The skin layer is simply the molecular layer that interfaces between a turbulent ocean and a turbulent atmosphere. The temperature difference between the skin and the bulk temperatures is directly related to the wind speed and net air–sea heat flux. Thus, an improved understanding of this relationship can help us to better resolve the net heat and momentum fluxes between the ocean and the atmosphere. In part, this understanding and the shift to the computation of skin SST from satellite infrared data depend on the creation of a network of ‘ship of opportunity’ based skin SST radiometers collecting global and continuous samples of skin SST ‘ground truth’ data (i.e., without an intervening atmosphere).

3.2 SCOPE:

This model will help us to increase the accuracy of the program which results in giving the better output. In this algorithm the null values will be taken as mean or median values where this is not present in previous versions. This surely helpful for investors for investing in Sea surface temperature.

3.2.1ADVANTAGES:

- Time saving as it is operated online.
- All the data can be collected from online itself.
- It can hold up on a huge number of data.
- Easy to understand.

3.3 SEA SURFACE TEMPERATURE ANALYSIS:

This study aims to forecast the movements of Sea surface temperature temperature at high degree of accuracy. To this end, machine learning (henceforth, ML) algorithms are applied, which do not require strict assumptions like traditional methods, (e.g., regression analysis, discriminant analysis, cluster analysis, etc.). While traditional models use whole data to investigate causal relations, ML algorithms normally split the data set into training and testing sets. Hence, ML allows computers to “learn” and make predictions. Although both methods try to increase the accuracy by minimizing some loss functions, ML does so use nonlinear algorithms. This does not mean that ML algorithms always outperform traditional models. However, ML algorithms, specially developed to address specific problems, may provide better forecasts for large data sets. All these make ML algorithms very popular for the scholars to apply. Many studies empirically compare traditional models and ML algorithms concerning their forecasting performances. In some studies, traditional models outperform ML algorithms, while in others the latter

outperform the former. So we use SVM,KNN,Decision Tree,Logistic regression algorithms to the data and analyses and predict the data.

CHAPTER 4

SOFTWARE DEVELOPMENT AND METHODOLOGY

4.1 DESCRIPTION OF DIAGRAM

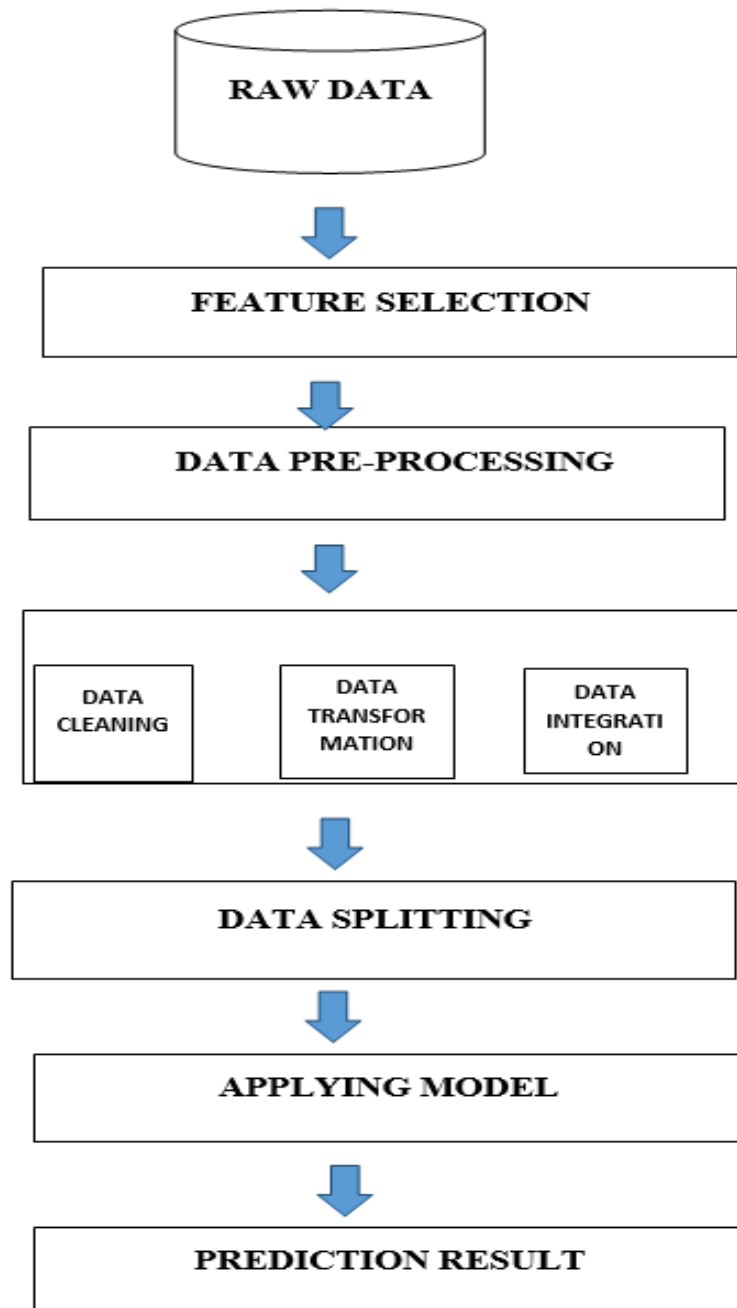


Fig 4.1.1 System Design

As by observing the above fig we can understand that it is a step by step process where one can say that it is a sequential process.

Raw Data -

As in the first step it is raw data where all the data is collected from the various resources. In this work, the data is collected from the last ten years with parameters such as (date, temperature, Co2 emission). There are many more websites to collect the data but these are the data collected to this project.

Feature Selection -

And the next step is about feature selection where as only the needed parameters are considered and taken into consideration so that it will be efficient for the program. As taking the raw data it consists of date, temperature, Co2 emission etc., by taking all set of columns decreases the accuracy of the program. So, without effecting the program main things such as date and temperature are taken into consideration so that the accuracy will not be affected.

Data Pre-processing -

When we talk about data, we usually think of some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc.

Machines don't understand free text, image or video data as it is, they understand 1s and 0s. So it probably won't be good enough if we put on a slideshow of all our images and expect our machine learning model to get trained just by that. In any Machine Learning process, Data Pre-processing is that step in which the data gets transformed, or Encoded, to bring it to such a

state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm.

In this process all the missing cases are combined together when and wherever possible. In this regression models the data set is splitted into a ratio of 70:30 where the 70% is trained data set and remaining 30% is validation set. So by using machine learning algorithm we perform the training for 70% of the data set and then the 30% data is validated on the trained data. It has further more steps where one by one is done in a sequential way Data Cleaning, Data Transformation, Data Integration

Data Cleaning-

In this method all the inaccurate data is corrected for all the record set..

Data cleaning is a technique that can remove unnecessary data such as null values.., null valued data causes model may be inaccurate and miss leading of model prediction

```
print df['temperature'].isnull()
```

Above code shows the null values of the particular column

Filling the null values can be done by as follows

```
df['temperature'].median()
```

```
df['temperature'].fillna(median, inplace=True)
```

Data Integration -

In this data is taken from the year 1995-2010 from website kaggle.com. And then the data is to be integrated into a single file. So that data integration is used in that way.

Data Splitting -

It splits the data into the two parts. which one part is for testing and another for training data with ratio 70%:30%

Applying Model –

Now the data is collected and all the modelling techniques are applied to the data. Support Vector Machine, K-Neighbors algorithms is applied to the data-set where 70% data is trained and remaining 30% data is tested for getting the resultant output.

Prediction Result -

Now the data is tested for 30% and checked with the trained data values. So by checking it can be able to predict the accuracy of the program.

4.2 SOFTWARE ENVIRONMENT-

PYTHON

Python is a high level and effective general use programming language. It supports multi-paradigms. Python has a large standard library which provide tools suited to perform various tasks. Python is a simple, less-clustered language with extensive features and libraries. Different programming abilities are utilized for performing the experiment in our work. In this thesis, the following python libraries were used.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of Python's other implementations. Python and CPython are managed by the non-profit Python Software Foundation.

It is used for:

1. web development (server-side),
2. software development,
3. mathematics,
4. system scripting.

DATASET

A Dataset is the basic data container in PyMVPA. It serves as the primary form of data storage, but also as a common container for results returned by most algorithms. The dataset assumes that the first axis of the data is to be used to define individual samples, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object

To train the model to predict the Sea surface temperature we need to have training data. Collect the data. The data might have some gaps.

Read the CSV files. The code is given below.

```
data = read.csv("../input/data.csv")
```

Remove the rows and columns that you will not use or which are not required.

Now, split the dataset into two parts. One becomes a training data set and another is test data set. Here we have taken 70% of the data set for training the model and the rest of the 30% are for test set.

```
n_train_rows = int(dataset.shape[0]*.8)-1
```

```
train = dataset.iloc[:n_train_rows, :]
```

```
test = dataset.iloc[n_train_rows:, :]
```

We are using KNN algorithm here, so we need to organize our data in blocks. Our data is organized in 1 minute intervals. So we will be using 50 blocks to predict.

Now prepare the training data.

```
x_train = []
```

```
y_train = []
```

```
for i in range(steps, training_set_scaled.shape[0]-steps):
```

```
x_train.append(training_set_scaled[i-steps:i, :])
```

```
y_train.append(training_set_scaled[i, :])
```

```
x_train, y_train = np.array(x_train), np.array(y_train)
```

```
print(x_train.shape)
```

Create a variable named `prediction_days`. This will be an array. Set this equal to the last 30 days of the original data.

Following the above code, prepare test data.

Now plot the graph of test data and training data.

Now test the model accuracy.

Print the predicted values and actual values. Do this for the remaining next 30 days.

JUPYTER

Jupyter Notebook is built off of IPython, an interactive way of running Python code in the terminal using the REPL model (Read-Eval-Print-Loop). The IPython Kernel runs the computations and communicates with the Jupyter Notebook front-end interface. It also allows Jupyter Notebook to support multiple languages. Jupyter is a free, open-source, interactive web tool known as a computational notebook, which researchers can use to combine software code, computational output, explanatory text and multimedia resources in a single document.

4.3 ALGORITHMS USED-

SVM (SUPPORT VECTOR MACHINE)-

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

The basics of Support Vector Machines and how it works are best understood with a simple example. Let's imagine we have two tags: red and blue, and our data has two features: x and y . We want a classifier that, given a pair of (x,y) coordinates, outputs if it's either red or blue. We plot our already labeled

training data on a plane.

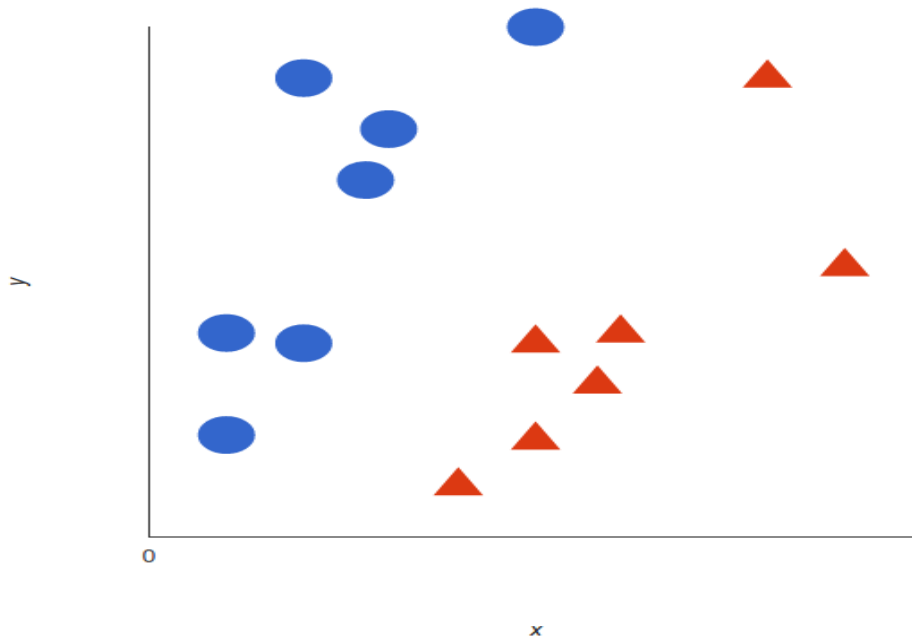


Fig 4.3.1 Our labeled data

A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it's simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

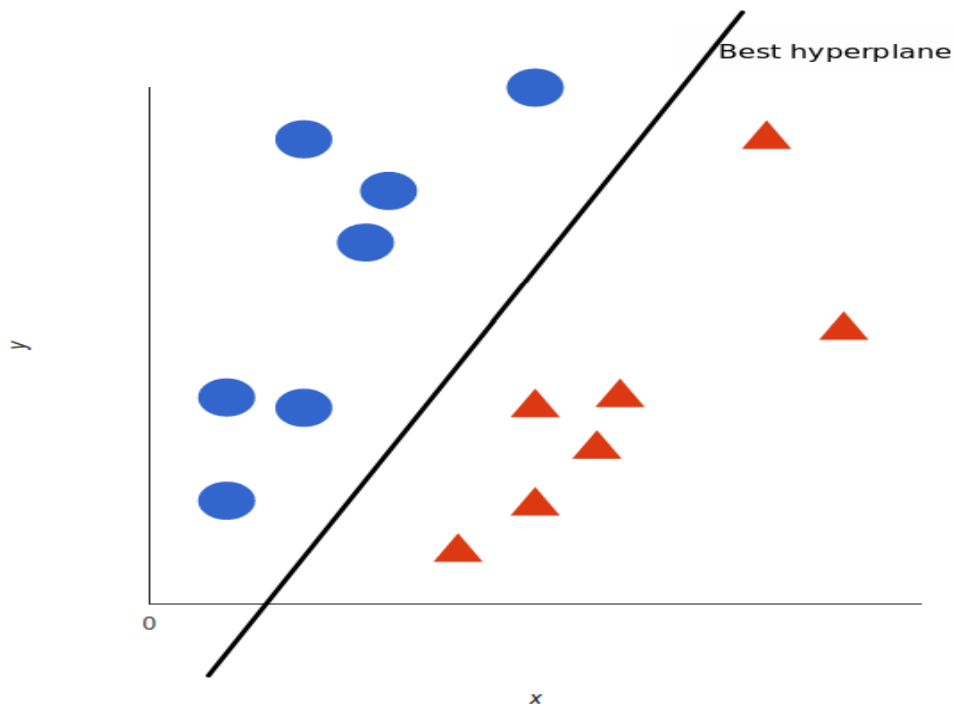


Fig 4.3.2 In 2D, the best hyperplane is simply a line

But, what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

Nonlinear data-

Now this example was easy, since clearly the data was linearly separable — we could draw a straight line to separate red and blue. Sadly, usually things aren't that simple.

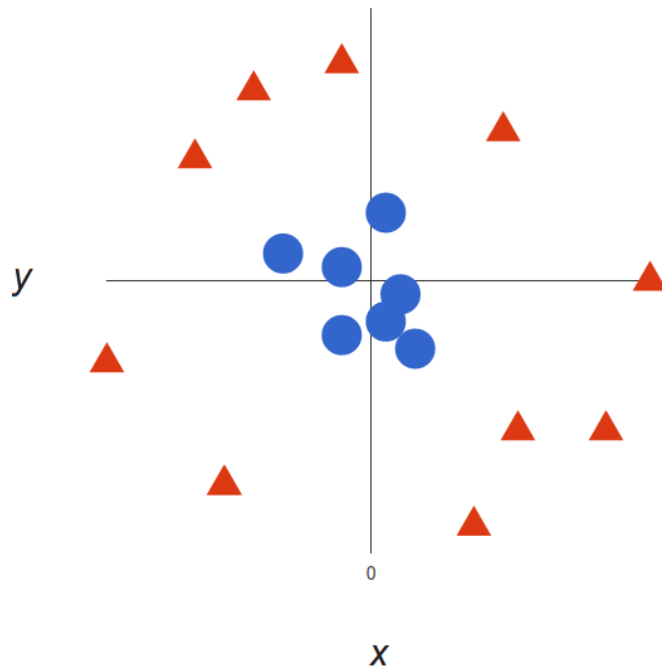


Fig 4.3.3 A more complex dataset

It's pretty clear that there's not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them.

So here's what we'll do: we will add a third dimension. Up until now we had two dimensions: x and y. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$ (you'll notice that's the equation for a circle).

This will give us a three-dimensional space. Taking a slice of that space, it looks like this:

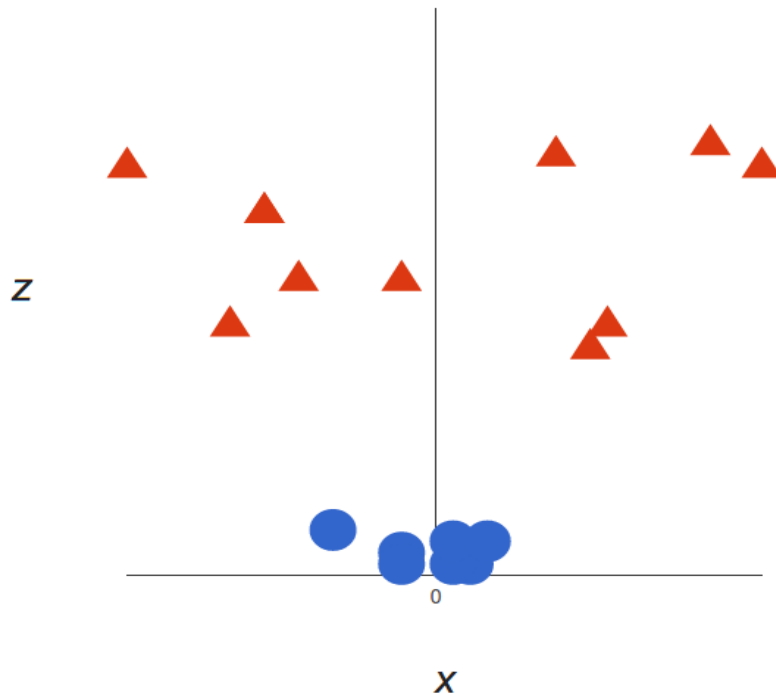


Fig 4.3.4 Before Division

From a different perspective, the data is now in two linearly separated groups

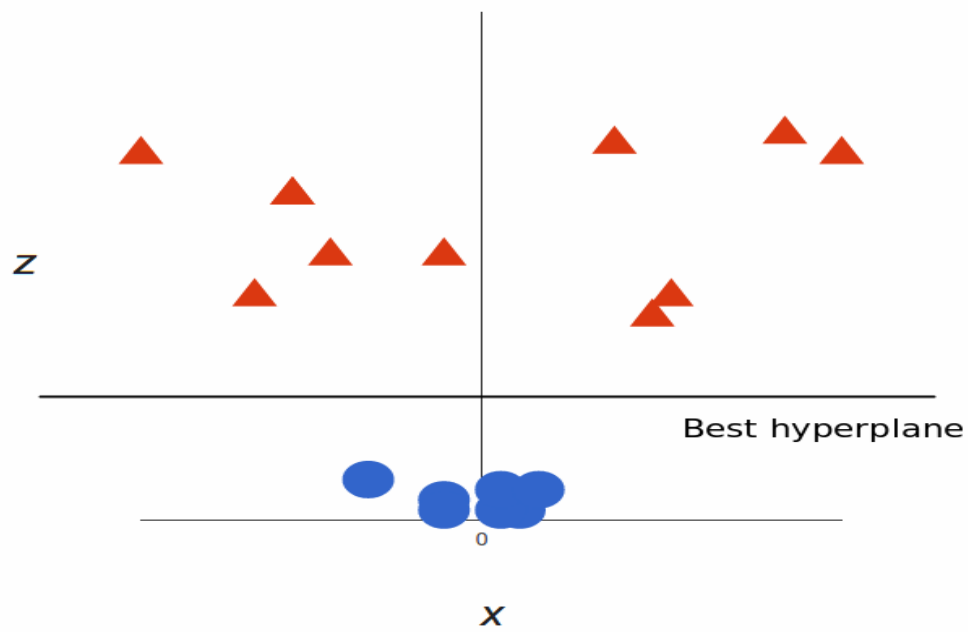


Fig 4.3.5 After Division

That's great! Note that since we are in three dimensions now, the hyperplane is a plane parallel to the x axis at a certain z (let's say $z = 1$).

What's left is mapping it back to two dimensions:

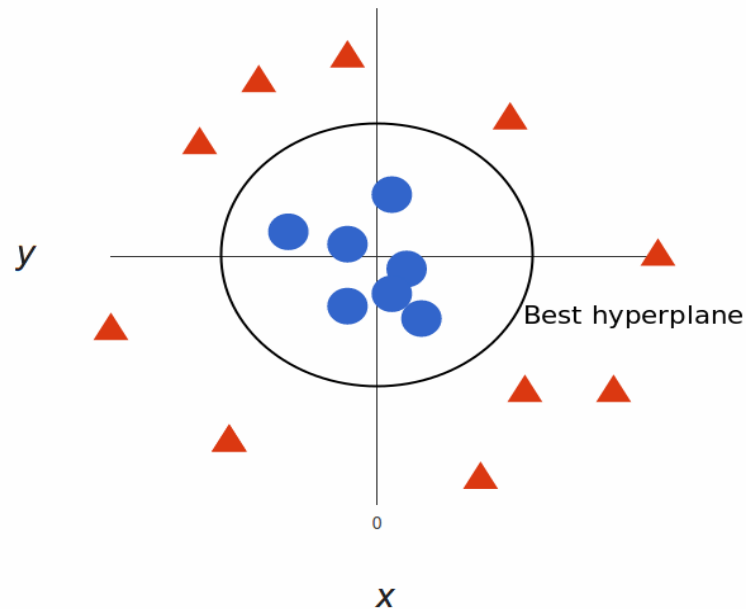


Fig 4.3.6 Back to our original view, everything is now neatly separated

And there we go! Our decision boundary is a circumference of radius 1, which separates both tags using SVM. Check out this 3d visualization to see another example of the same effect:

And in this SVM algorithm we also have kernels

Kernel-

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (x_i) is calculated as follows:

$$f(x) = B(0) + \sum(a_i * (x, x_i))$$

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

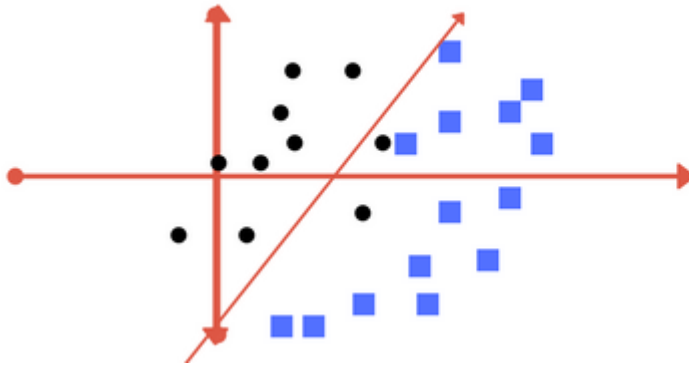
The polynomial kernel can be written as $K(x, x_i) = 1 + \sum(x * x_i)^d$ and exponential as $K(x, x_i) = \exp(-\gamma * \sum((x - x_i)^2))$. *Polynomial and exponential kernels calculates separation line in higher dimension. This is called kernel trick*

Regularization

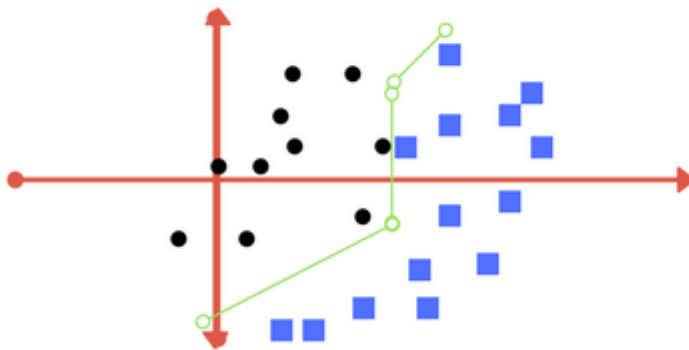
The Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

The images below (same as image 1 and image 2 in section 2) are example of two different regularization parameter. Left one has some misclassification due to lower regularization value. Higher value leads to results like right one.



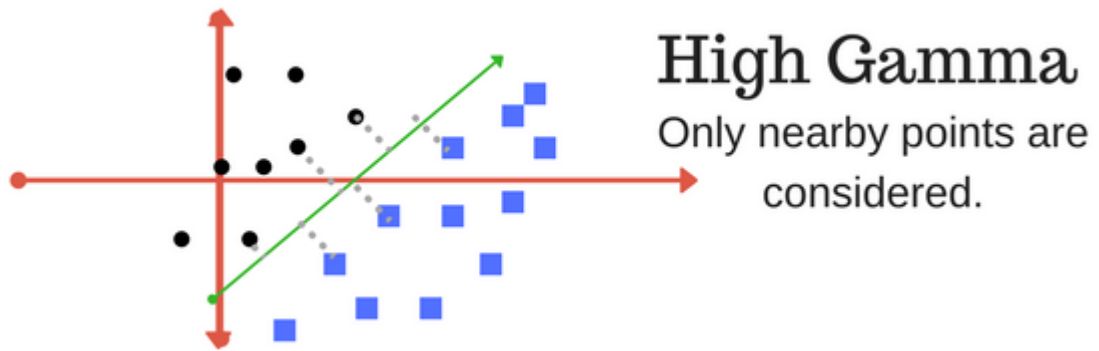
4.3.7 Low regularization value



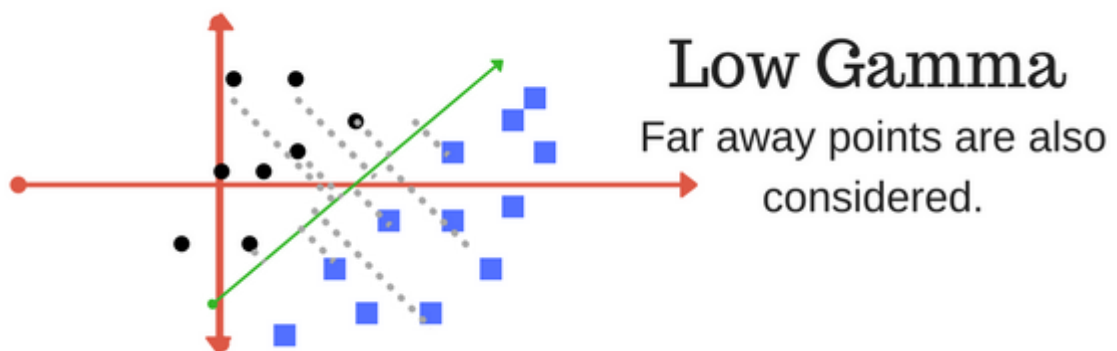
4.3.8 High regularization value

Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible line are considered in calculation.



4.3.9 High Gamma

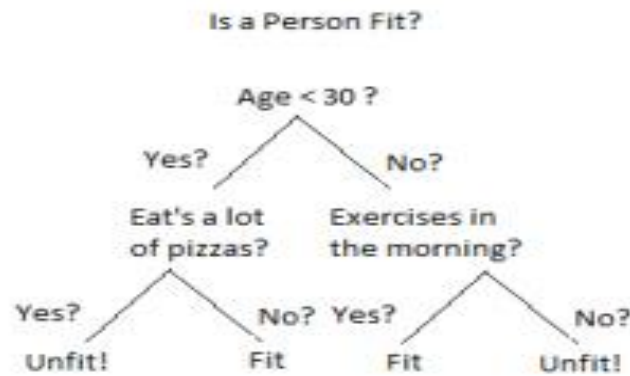


4.3.10 Low Gamma

Decision Trees for Classification Algorithm:

Introduction

Decision Trees are a type of Supervised Machine Learning (that is you explain what the input is and what the corresponding output is in the training data) where the data is continuously split according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.



An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity, etc. The decision nodes here are questions like 'What's the age?', 'Does he exercise?', 'Does he eat a lot of pizzas'? And the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes no type problem).

There are two main types of Decision Trees:

Classification trees (Yes/No types)

What we've seen above is an example of classification tree, where the outcome was a variable like 'fit' or 'unfit'. Here the decision variable is Categorical.

Regression trees (Continuous data types)

Here the decision or the outcome variable is Continuous, e.g. a number like 123.

Working

Now that we know what a Decision Tree is, we'll see how it works internally. There are many algorithms out there which construct Decision Trees, but one of the best is called as ID3 Algorithm. ID3 Stands for Iterative Dichotomiser 3.

Before discussing the ID3 algorithm, we'll go through few definitions.

K-NEAREST NEIGHBOR(KNN) CLASSIFICATION METHOD

K-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms. The neighbors are taken from a set of objects for which the class (for k-NN classification) or the object property value (for k-NN regression) is known.

STEP 1: BEGIN

STEP 2: Input: $D = \{(x_1, c_1), \dots, (x_N, c_N)\}$

STEP 3: $x = (x_1, \dots, x_n)$ new instance to be classified

STEP 4: FOR each labelled instance (x_i, c_i) calculate $d(x_i, x)$

STEP 5: Order $d(x_i, x)$ from lowest to highest, $(i = 1, \dots, N)$

STEP 6: Select the K nearest instances to x: D_x^k

STEP 7: Assign to x the most frequent class in D_x^k

STEP 8: END

4.4 MODULES USED

NumPy-

NumPy is a Python library used for working with arrays.

It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.

NumPy stands for Numerical Python.

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Pandas-

Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in any language. It is already well on its way toward this goal.

pandas is well suited for many different kinds of data:

- Tabular data with heterogeneously-typed columns, as in an SQL table or Excel spreadsheet
- Ordered and un ordered (not necessarily fixed-frequency) time series data.
- Arbitrary matrix data (homogeneously typed or heterogeneous) with row and column labels
- Any other form of observational / statistical data sets. The data need not be labeled at all to be placed into a pandas data structure

The two primary data structures of pandas, Series (1-dimensional) and Data Frame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering. For R users, Data Frame provides everything that R's data. Frame provides and much more.

pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

Here are just a few of the things that pandas does well:

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating-point data
- Size mutability: columns can be inserted and deleted from Data Frame and higher dimensional objects
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, Data Frame, etc. automatically align the data for you in computations
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into Data Frame objects
- Intelligent label-based slicing, fancy indexing, and sub setting of large data sets
- Intuitive merging and joining data sets.

Scikit-learn-

Scikit-learn (formerly skits. learn and also known as sklearn) is a free software machine learning library for the Python programming language.[3] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k -means and DBSCAN, and is designed to inter operate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cpython to improve performance. Support vector machines are implemented by a Cpython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas data frames, scipy, and many more.

4.5.ARCHITECTURAL DESIGN:

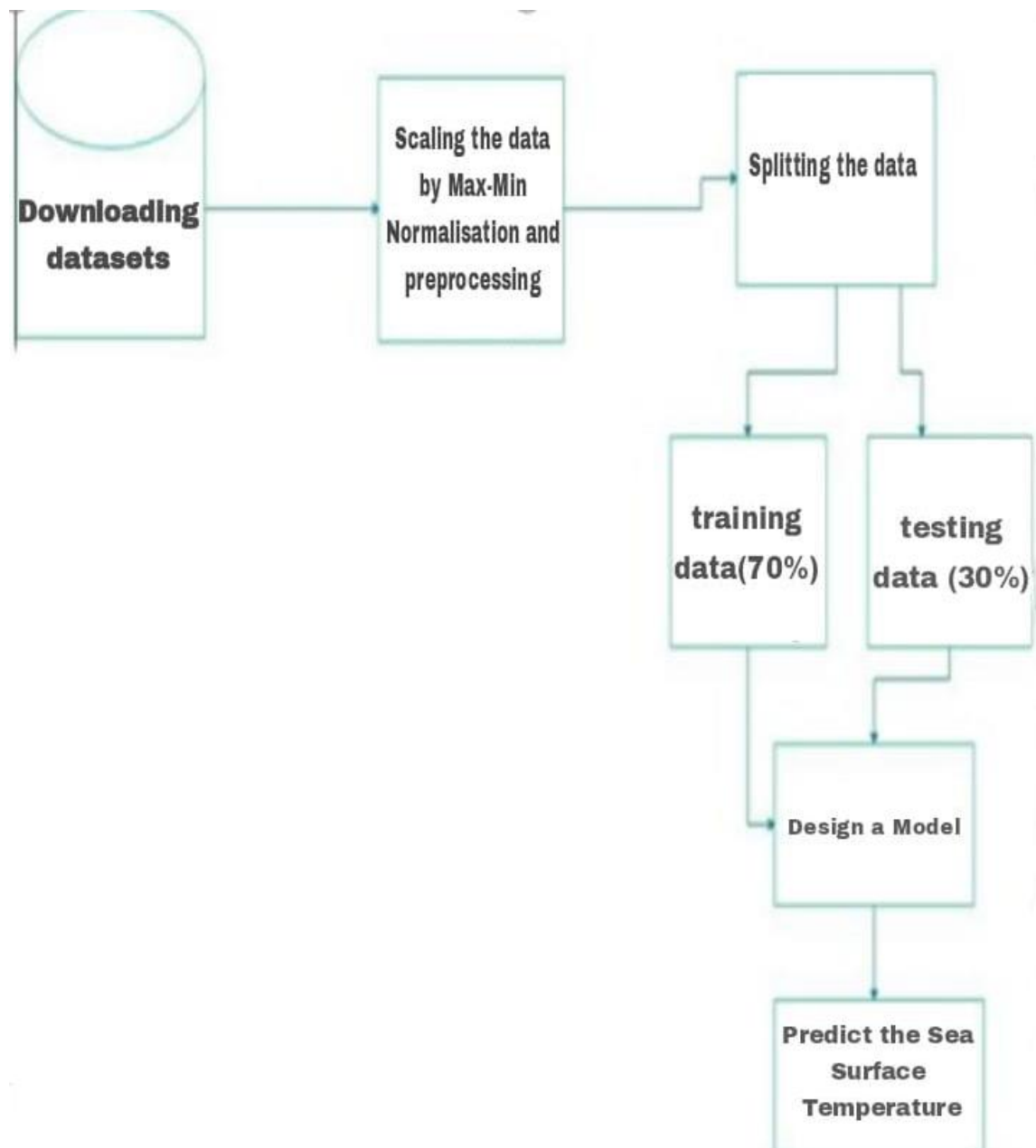


Fig 4.5 ARCHITECTURAL DESIGN

CHAPTER 5

RESULT AND DISCUSSION

5.1 EXPERIMENT RESULTS-

This chapter presents the results that are obtained from the experiment. In this work, have developed and compared various algorithms based on machine learning. Convolutional Neural Networks and combined both and also regression type but that algorithms are much more effective in other algorithms ,even in sequence data analysis but not in sea surface temperature analysis and prediction , why because when work did with random forest ,first it will convert in decision tree it will take high time to train data, in the same if work did in support vector machine (SVM) in this algorithm the main advantage is keras function .By using this function it is easy to calculate or manipulating the number functions, even it will take less time to train data when compared to other function and another advantage is time complexity is low when compared to Convolutional neural network and random forest, next about space complexity even in this category Support Vector Machine is very good. Finally if go to the accuracy, when work use support vector machine algorithm it was getting high when compared to other any algorithms so, in every case and support vector machine is very good and user friendly and even easy to understand .The data that has been used for the analysis has been taken up from station 46012 of the National Data Buoy Centre. The data set that has been used ranges for about 27 years from 1980 to 2007 with a total of 177549 data entries. The data is highly quality controlled and gives us the true measure of the sea temperature. Table 1 portrays the average monthly sea temperature.

Support vector machine is the classifier that maximizes the margin. The goal of a classifier in our example below is to find a line or $(n-1)$ dimension hyper-plane that separates the two classes present in the n -dimensional space. The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of the classifier. These points are called the support vectors. They

support or define the hyperplane. Supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labelled training data for each category, they're able to categorize new text. So you're working on a text classification problem. The support vector uses a mathematical function, often called a kernel function which is a math function that matches the new data to the best image from the training data in order to predict the unknown picture's label

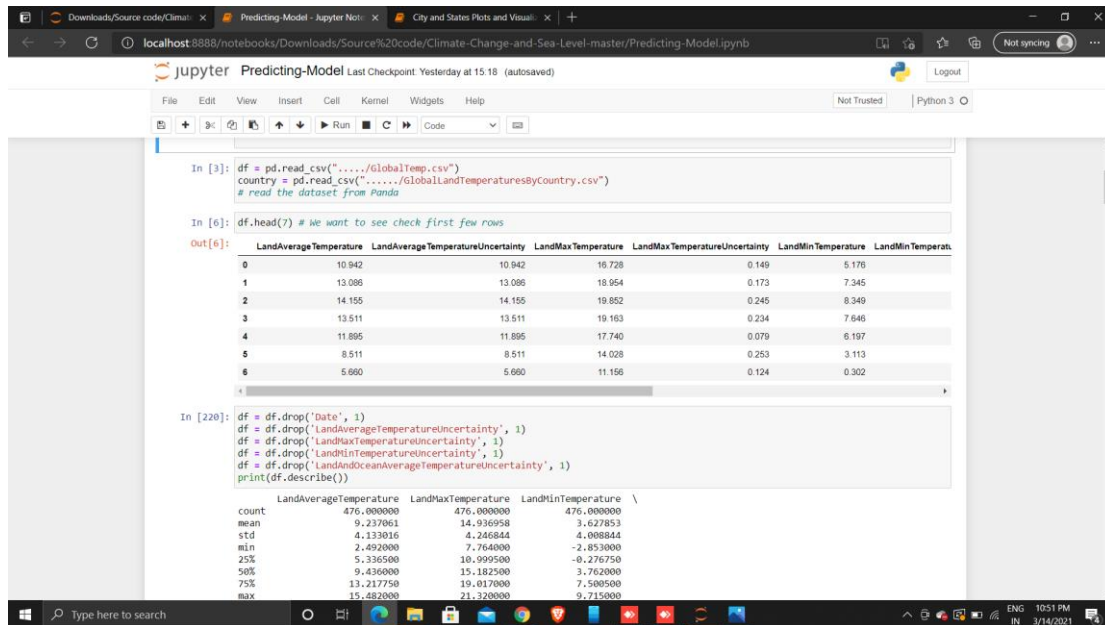


Fig 5.1 Dataset

So by observing the above figure5.1 at the starting of the program all the libraries are imported and dataset is collected. Then variables are taken and

trained.

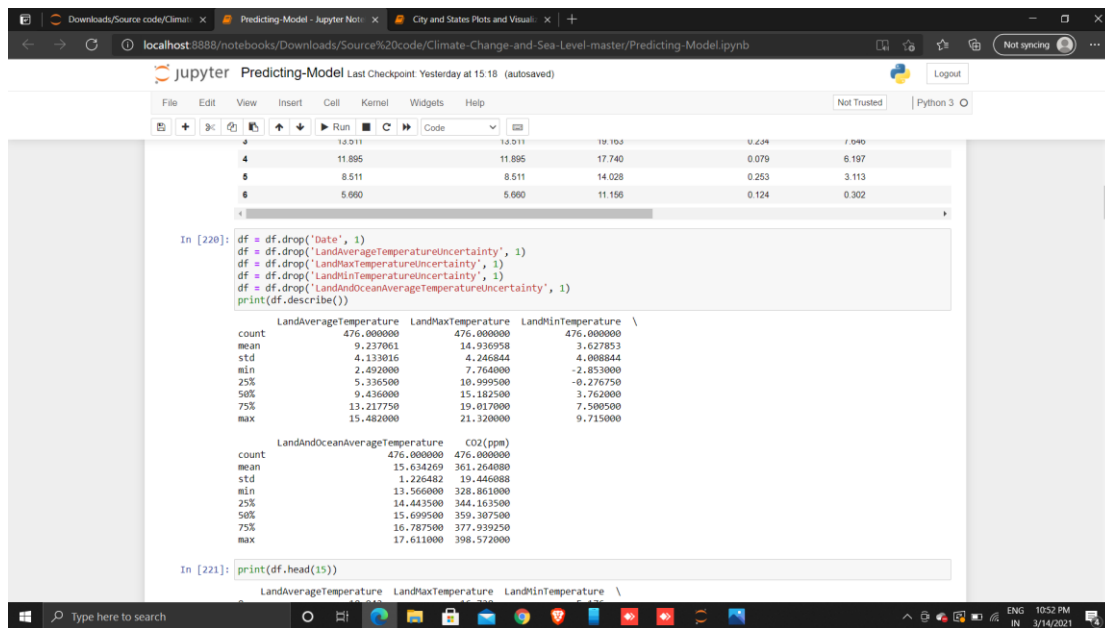


Fig 5.2 Trained Dataset

By observing the above fig5.2 svm is imported and all the modules are taken such as sklearn and training and testing is performed on the data set. At the end of the program is the value which is obtained by training.

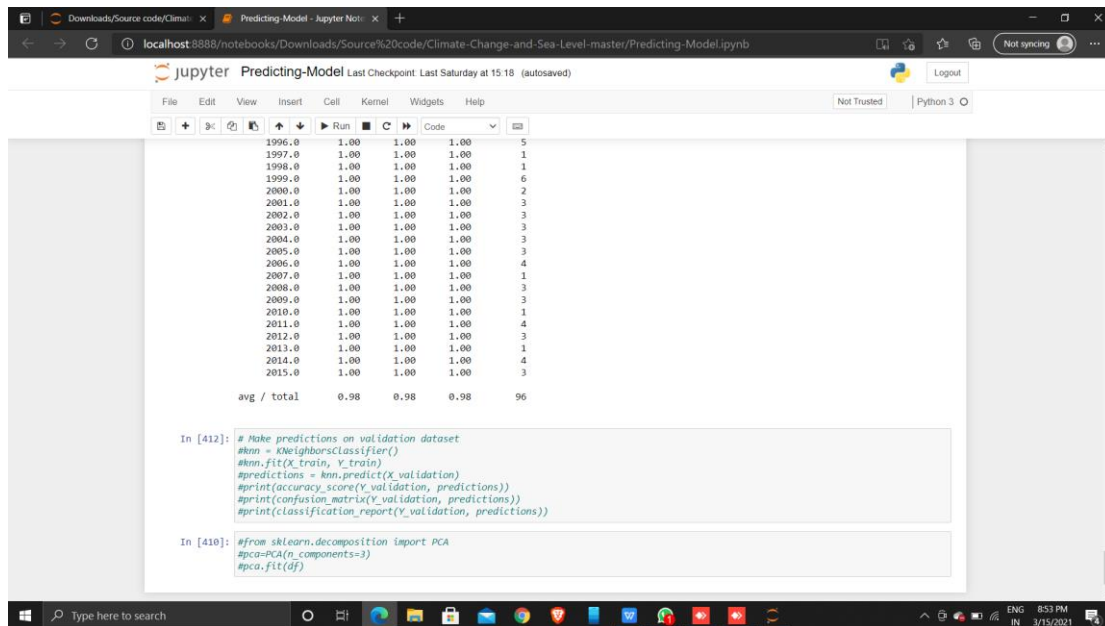


Fig 5.3 Output

So above shown fig 5.3 is the output in which temperature is predicted for the next coming days.

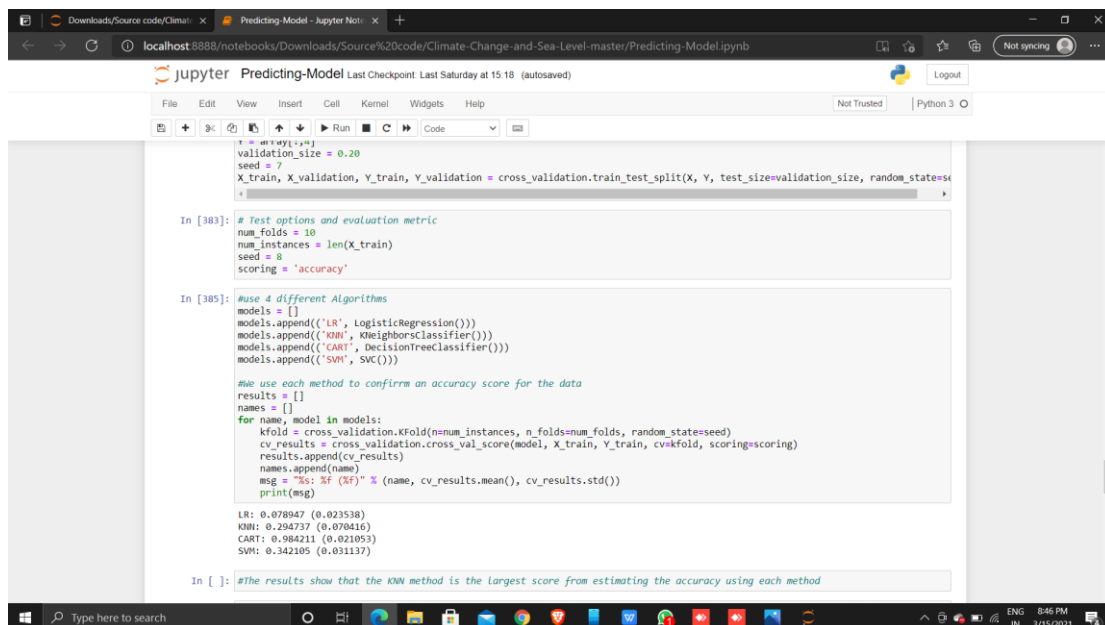


Fig 5.4 Accuracy

So above fig gives the accuracy of the program which is 65% which is far better than all the other algorithms which are used preciously for temperature

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 CONCLUSION-

The importance and the scope of the analysis of sea surface temperature is a highly important aspect of being able to monitor and mitigate the vagaries of global warming as well as in ensuring a smooth transition to using renewable energy in the form of hydro generated energy. This review represented a basic outlook, of how average temperature data of a station 46012 of the National Data Buoy Centre fitted with respect to basic distribution functions. This review can be furthered upon by using other distributions such as mixture distributions to get better fits. Even regression models which is very popularly used in case for modelling solar insolation and rainfall may be used for modelling the sea surface temperature.

6.2 FUTURE WORK-

There are several research directions for future work. First, it would be interesting to consider other recent deep learning models, such as transformer networks and few-shot/one-shot learning, to build prediction models. But this model has disadvantages also because of the

REFERENCES

- [1] Rao, Sethuraman N., "A novel solution for high speed internet over the oceans," IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2018.
- [2] Rao, Sethuraman N., Maneesha Vinodini Ramesh, and Venkat Rangan. "Mobile infrastructure for coastal region offshore communications and networks." In Global Humanitarian Technology Conference (GHTC), 2016, pp. 99-104. IEEE, 2016.

[3] Karthik, A., Deepu George Koshy, Lakshmi Rajagopal, Anjana Luke, M. S. Meera, and NB Sai Shibu, "Study and analysis of OceanNet—Marine internet service for fishermen," In Global Humanitarian Technology Conference (GHTC), 2017 IEEE, pp. 1-8. IEEE, 2017.

[4] Vijayan, Deepthy M., and Sreedevi K. Menon, "Direction of arrival estimation in smart antenna for marine communication," In Communication and Signal Processing (ICCSP), 2016 International Conference on, pp. 1535-1540. IEEE, 2016.

[5] C.J. Donlon , "IR radiometers,"1st edition of Encyclopedia of Ocean Sciences (2001) Vol 3: 1374–1385.

SOURCE CODE-

```
import pandas as pd

import numpy as np

import matplotlib as plt

import sys

import scipy

import sklearn

import random


from pandas.tools.plotting import scatter_matrix

import matplotlib.pyplot as plt


from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

#from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

from sklearn.svm import SVC

from sklearn.linear_model import LogisticRegression

from sklearn import cross_validation

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier
```

```

models = []

models.append(('LR', LogisticRegression()))

models.append(('KNN', KNeighborsClassifier()))

models.append(('CART', DecisionTreeClassifier()))

models.append(('SVM', SVC()))


#We use each method to confirm an accuracy score for the data

results = []

names = []

for name, model in models:

    kfold = cross_validation.KFold(n=num_instances, n_folds=num_folds,
    random_state=seed)

    cv_results = cross_validation.cross_val_score(model, X_train, Y_train,
    cv=kfold, scoring=scoring)

    results.append(cv_results)

    names.append(name)

    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())

    print(msg)

Compare Algorithms

fig = plt.figure()

fig.suptitle('Algorithm Results (LR , KNN, CART, SVM)')

#ax = fig.add_subplot(111)

plt.boxplot(results)

```

```
ax.set_xticklabels(names)
```

```
plt.show()
```

SCREEN SHOTS-

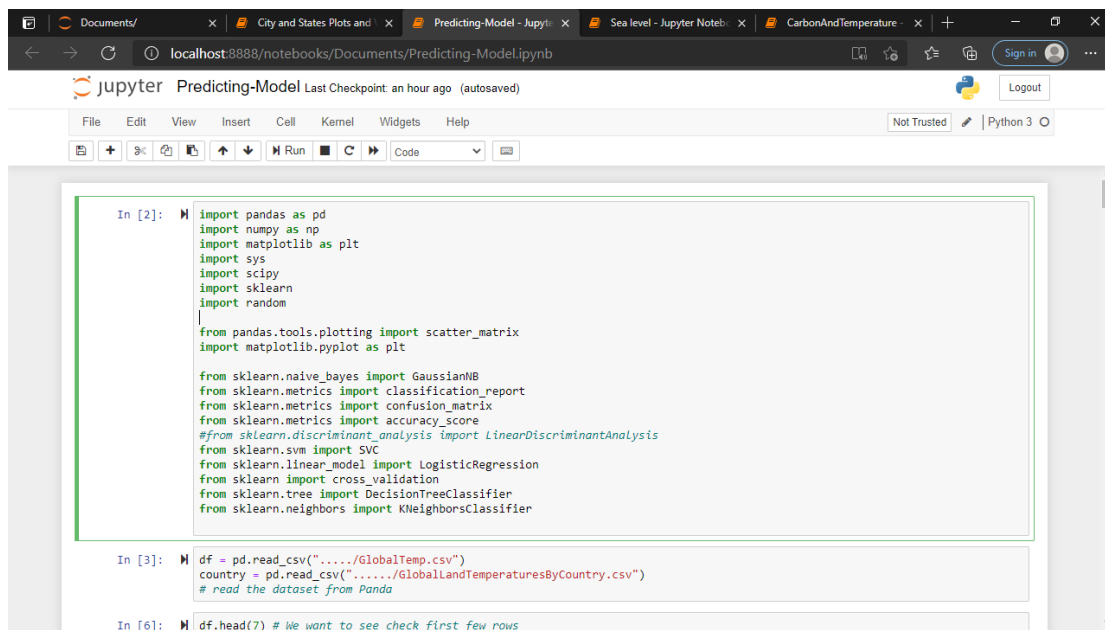


Fig 6.1 Importing libraries

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [3]: df = pd.read_csv("...../GlobalTemp.csv")
country = pd.read_csv("...../GlobalLandTemperaturesByCountry.csv")
# read the dataset from Panda
```

```
In [6]: df.head(7) # We want to see check first few rows
```

```
Out[6]:
```

	LandAverageTemperature	LandAverageTemperatureUncertainty	LandMaxTemperature	LandMaxTemperatureUncertainty	LandMinTemperature	LandMinTemp
0	10.942	10.942	16.728	0.149	5.176	
1	13.086	13.086	18.954	0.173	7.345	
2	14.155	14.155	19.852	0.245	8.349	
3	13.511	13.511	19.163	0.234	7.646	
4	11.895	11.895	17.740	0.079	6.197	
5	8.511	8.511	14.028	0.253	3.113	
6	5.660	5.660	11.155	0.124	0.302	

```
In [220]: df = df.drop('Date', 1)
df = df.drop('LandAverageTemperatureUncertainty', 1)
df = df.drop('LandMaxTemperatureUncertainty', 1)
df = df.drop('LandMinTemperatureUncertainty', 1)
df = df.drop('LandAndOceanAverageTemperatureUncertainty', 1)
print(df.describe())
```

The output of the last cell shows the columns: LandAverageTemperature, LandMaxTemperature, LandMinTemperature, \

Fig 6.2 Performing Operation

The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [252]: df.describe(percentiles=[.05,.25,.75,.95])
```

```
Out[252]:
```

	LandAverageTemperature	LandMaxTemperature	LandMinTemperature	LandAndOceanAverageTemperature	CO2(ppm)
count	473.000000	473.000000	473.000000	473.000000	473.000000
mean	9.214922	14.914288	3.606740	15.629645	361.467653
std	4.135350	4.249428	4.011353	1.228473	19.338140
min	2.492000	7.764000	-2.853000	13.566000	330.529000
5%	3.284400	8.713600	-2.076200	13.847600	333.682400
25%	5.245000	10.725000	-0.285000	14.436000	344.236000
50%	9.378000	15.170000	3.697000	15.675000	359.442000
75%	13.217000	19.006000	7.482000	16.786000	377.979000
95%	14.835400	20.652400	9.132600	17.406000	393.869600
max	15.482000	21.320000	9.715000	17.611000	398.572000

```
In [374]: names = ['LandAverageTemperature', 'LandMaxTemperature', 'LandMinTemperature', 'CO2', 'Date']
df2= pd.read_csv(temp, names=names)
#had to use another method to verify information by deleting labels
```

```
In [375]: print(df2.describe())
```

Fig 6.3 Taking variables

```

In [385]: #use 4 different Algorithms
models = []
models.append(('LR', LogisticRegression()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('SVM', SVC()))

#We use each method to confirm an accuracy score for the data
results = []
names = []
for name, model in models:
    kfold = cross_validation.KFold(n=num_instances, n_folds=num_folds, random_state=seed)
    cv_results = cross_validation.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)

LR: 0.078947 (0.023538)
KNN: 0.294737 (0.070416)
CART: 0.984211 (0.021053)
SVM: 0.342105 (0.031137)

In [ ]: #The results show that the KNN method is the largest score from estimating the accuracy using each method

In [402]: # Compare Algorithms
fig = plt.figure()
fig.suptitle('Algorithm Results (LR , KNN, CART, SVM)')
#ax = fig.add_subplot(111)
plt.boxplot(results)

```

Fig 6.4 Using Algorithms

1970	18.794121	0.341842	15612006.0
1971	18.588207	0.334640	15761465.0
1972	18.605612	0.352646	16542539.0
1973	18.873293	0.356324	17514128.0
1974	18.631447	0.337361	17456557.0

```

In [105]: from sklearn.linear_model import LinearRegression
#use 22 years for train data and next 22 as test
train = df[:24]
test = df[23:]
X_train = train.drop('Average Temp', axis=1).reset_index().drop('index', axis=1)
X_test = test.drop('Average Temp', axis=1).reset_index().drop('index', axis=1)
Y_train = train['Average Temp']
Y_test = test['Average Temp']
lr_elect = LinearRegression()
lr_elect.fit(X_train, Y_train)
lr_elect.score(X_test, Y_test)

Out[105]: 0.14408479188974976

In [106]: y_lr = lr_elect.predict(X_test)
plt.figure(figsize=(12,6))
plt.scatter(X_test.index, Y_test, label='Observed', color='k')
plt.plot(X_test.index, y_lr, label='Predicted', color='g')
plt.legend(loc='upper right')

Out[106]: <matplotlib.legend.Legend at 0x115104470>

```

Fig 6.5 Predicting x variables

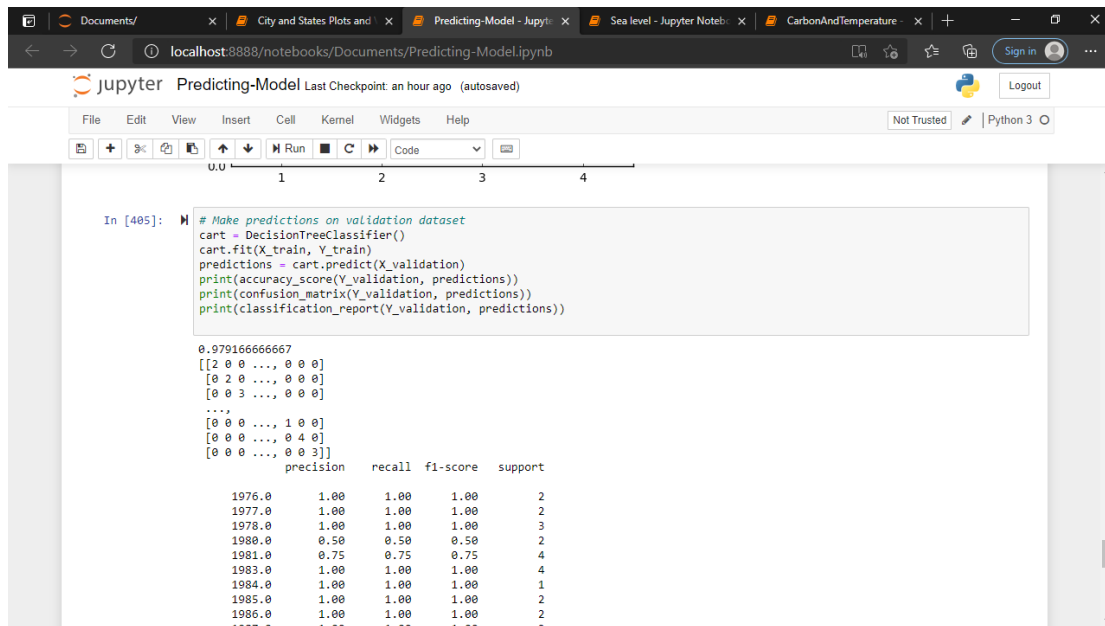


Fig 6.6 Result

Plagiarism report

