

Deccan Education Society's
Fergusson College (Autonomous), Pune
Department of Computer Science

A

**Report
on**

***“Weather Predictor using
KNN Algorithm”***

In partial fulfillment of Post Graduate course
in

M.Sc. Computer Science – I
(Semester -II)

CSC-570 Practical III

SUBMITTED BY

Akil shaikh (236248)
Adesh jadhav (236230)
Dagale Mangesh (236206)

Index

Sr. No	Table of Content	Page No
1	Case Study description (Algorithm details and UI design details)	
2	Screenshots	
3	References (if you have used any)	

1. Case Study Description:

❖ Introduction:

We are building a weather prediction application using the k-Nearest Neighbors (k-NN) algorithm. The application allows users to input weather parameters for a specific city and predicts the weather condition based on the k-NN algorithm.

❖ Project Overview:

- The project involves building a React application that utilizes the k-NN algorithm to predict weather conditions.
- Users can input parameters such as temperature, humidity, and wind speed for a particular city.
- The application then calculates the nearest neighbors based on the input parameters and predicts the weather condition.

❖ Technology :

- The application is built using React, a popular JavaScript library for building user interfaces.
- It also involves CSS for styling components and providing a user-friendly interface.

❖ Algorithm Description:

The k-Nearest Neighbors (k-NN) algorithm is a versatile and intuitive machine learning technique used for both classification and regression tasks. Here's a more detailed breakdown of how the algorithm works:

- **Initialization:**

- The algorithm begins by loading the dataset containing examples with their corresponding labels. Each example represents a data point in a multidimensional feature space.

- **Choose the Number of Neighbors (k):**

- The value of k is a crucial parameter in k-NN, representing the number of nearest neighbors to consider when making predictions.
- The choice of k influences the bias-variance tradeoff. Smaller values of k lead to more flexible models with high variance but low bias, while larger values of k result in smoother decision boundaries with low variance but high bias.

- **Calculate Distance:**

- For each query instance, the algorithm calculates the distance between the query instance and all the instances in the dataset.
- The distance metric used is typically Euclidean distance, although other distance metrics such as Manhattan distance or Minkowski distance can also be employed depending on the nature of the data.

- **Find Nearest Neighbors:**

- Once distances are calculated, the algorithm identifies the k instances with the smallest distances to the query instance.
- These instances constitute the nearest neighbors of the query instance and are used to make predictions.
- Determine Majority Vote (Classification) / Calculate Average (Regression):
- For classification problems, the algorithm determines the class labels of the k nearest neighbors and performs a majority vote to decide the class label of the query instance.
- In regression problems, the algorithm calculates the average of the target values (e.g., temperature, humidity) of the k nearest neighbors.

- **Output Result:**

- Finally, the algorithm outputs the predicted class label (for classification) or the predicted value (for regression) for the query instance.

- **UI Design Details:**

- The UI includes input fields for users to enter the city name, temperature, humidity, and wind speed.
-
- There's a button to trigger the prediction process and display the predicted weather condition.
-
- Additional buttons allow users to navigate between pages, such as returning to the home page or viewing algorithm step

2. ScreenShots:

Home page:

The k-Nearest Neighbors algorithm (k-NN) is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. It's a lazy learning algorithm as it doesn't explicitly learn a model. Instead, it chooses to memorize the training instances which are then used as "knowledge" for the prediction phase.



How Does k-NN Work?

The k-NN algorithm works by finding the distances between a query and all the examples in the data, selecting the specified number examples (k) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Applications of k-NN

k-NN can be used in a variety of settings, including:

- Handwriting detection
- Weather prediction
- Image recognition
- Recommendation systems
- Search engines
- And many more areas

Implementation of K-NN

We made an simple application of Weather prediction, Here we implement K-NN and show how to actual this Algorithm Works.

[Open](#)

Output:

Weather App

Weather Prediction using KNN Algorithm

Enter city:

Enter Temperature °C:

Enter Humidity %:

Enter Wind Speed km/h:

[Show Result](#)

Today is LightRainfall in pune.

[Show steps](#)

[Home](#)

Steps:

k-NN Algorithm Steps

Step 1: Load the Data

Load the dataset containing examples with their corresponding labels.

Step 2: Choose the Number of Neighbors (k)

Determine the value of k, which represents the number of nearest neighbors to consider for classification or regression.

Step 3: Calculate Distance

Calculate the distance between the query instance and all the instances in the dataset using a distance metric such as Euclidean distance.

Step 4: Find Nearest Neighbors

Identify the k instances with the smallest distances to the query instance.

3. References:

1. <https://www.geeksforgeeks.org>
2. <https://www.w3schools.com>
3. <https://legacy.reactjs.org>