

Capstone Project Report

(Project term January-May 2023)

**Weather Prediction Using Machine Learning**

Submitted by

**Ajay Rawat Registration Number: 11909456**

**Adarsh Kumar Registration Number: 11909469**

**Project Group Number: CSERGC0383**

**Course Code CSE445**

Under the Guidance of

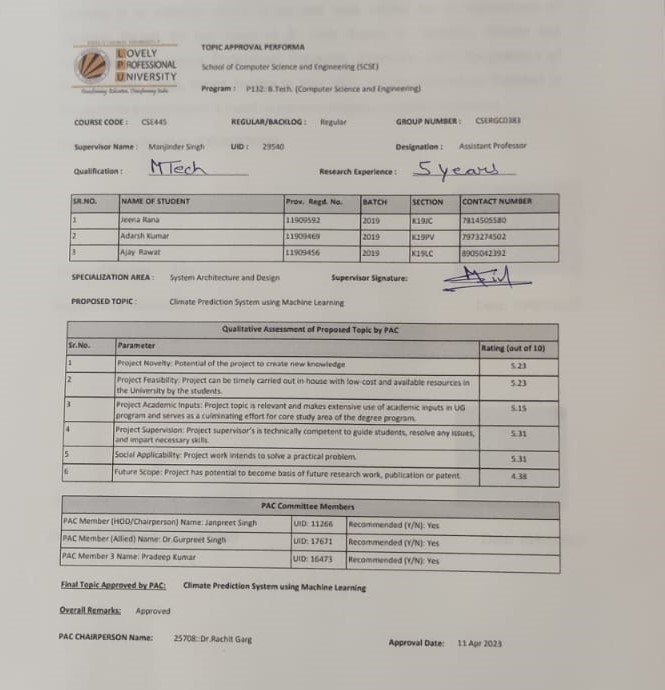
**Mr. Manjinder Singh Assistant Professor**

**School of Computer Science and Engineering**





PAC Form

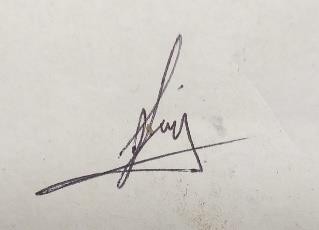


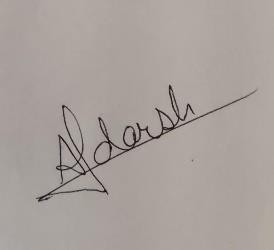


DECLARATION

We hereby declare that the project work entitled Weather Prediction Using Machine Learning is an authentic record of our own work carried out as requirements of Capstone Project for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Manjinder Singh, during January to May 2023. All the information furnished in this capstone project report is based on our own intensive workand is genuine.

Project Group Number: CSERGC0383

Name of Student 1: Ajay Rawat  Registration Number: 11909456 Date: 10/05/2023

Name of Student 2: Adarsh Kumar  Registration Number: 11909469 Date: 10/05/2023



Certificate

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort, and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfilment of the conditions for the award of B. Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

![Text, letter

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RpaRXhpZgAATU0AKgAAAAgABgALAAIAAAAmAAAIYgESAAMAAAABAAEAAAExAAIAAAAmAAAIiAEyAAIAAAAUAAAIrodpAAQAAAABAAAIwuocAAcAAAgMAAAAVgAAEUYc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFdpbmRvd3MgUGhvdG8gRWRpdG9yIDEwLjAuMTAwMTEuMTYzODQAV2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NAAyMDIzOjA1OjEwIDIyOjUxOjE4AAAGkAMAAgAAABQAABEckAQAAgAAABQAABEwkpEAAgAAAAM0MAAAkpIAAgAAAAM0MAAAoAEAAwAAAAEAAQAA6hwABwAACAwAAAkQAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMzowNToxMCAyMjo0ODoyNQAyMDIzOjA1OjEwIDIyOjQ4OjI1AAAAAAYBAwADAAAAAQAGAAABGgAFAAAAAQAAEZQBGwAFAAAAAQAAEZwBKAADAAAAAQACAAACAQAEAAAAAQAAEaQCAgAEAAAAAQAACK0AAAAAAAAAYAAAAAEAAABgAAAAAf/Y/9sAQwAIBgYHBgUIBwcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/9sAQwEJCQkMCwwYDQ0YMiEcITIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIy/8AAEQgAMAEAAwEhAAIRAQMRAf/EAB8AAAEFAQEBAQEBAAAAAAAAAAABAgMEBQYHCAkKC//EALUQAAIBAwMCBAMFBQQEAAABfQECAwAEEQUSITFBBhNRYQcicRQygZGhCCNCscEVUtHwJDNicoIJChYXGBkaJSYnKCkqNDU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6g4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2drh4uPk5ebn6Onq8fLz9PX29/j5+v/EAB8BAAMBAQEBAQEBAQEAAAAAAAABAgMEBQYHCAkKC//EALURAAIBAgQEAwQHBQQEAAECdwABAgMRBAUhMQYSQVEHYXETIjKBCBRCkaGxwQkjM1LwFWJy0QoWJDThJfEXGBkaJicoKSo1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoKDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uLj5OXm5+jp6vLz9PX29/j5+v/aAAwDAQACEQMRAD8A9HoqBiUYoATFLigBpWmkYNAxgKvnawO0kHB7jrQQKAEpMCgA4o2ikAu0UbaAFA4pcCgBMCjAoANo96TAFADdoo2igAwBSbRTANooxQAYFJikAhUU3aKAL9FMQUUANR1ddysGHsadQAlNNCAxtIlxdX8DYybgyqM/wsP8Qa1j0qpbgNNITUjDNKDzSAUUtABnFJn3oAXNGaADNIaAEzTc80AGaKACigBM0UAJRQBcopiGk00sfemgON0rVJ9I1TULa6AW0W455yYt3KsPVTzn8K6dNTWZM20Mk/oU4X/vo1pOACPJe+W0kslvbooLHq+B7ngVHJatcw5a+nZGXIMbhAQR2IqLAZ+kWwtZ4gXZ2NovzMck/M3U/jW32oluAhNNJqRjS5xVe31CC4uJ4I5Q0tu22Re4NO1wLYYmnEmkAmTSZoAUnim7uevQ880AOUk00scmgBMmkJNACbsdaVGoAkopAUdWeaPTLh4HKyhPkIHQ0/TboX2m29z/AM9EBPse/wCtVb3QLVJUgXKjlmjhQvK6og7saoRTfUCyEwx/L2lmby0/Xk1WEyTLiS8kmz1W1jbb+YH9apIDIe0ez8TW1xBp0r2txEYJQ65+bqCc5xxmrc0Vzopa7tbSGG1J/fxGYkc/xAAcYrS9wNOymbULabzo4miJKK0ZJWQY5IzRFPE2jCaFPLjEbBU/u4yMfpUMCJV8m6sQerRNEfqAG/oav8CpluBBLcQQDdLNHGPV3AqrHqtlNKscU4cscAqpK/n0pWuMtla4lFms/Gur31tHLKqNGJoYxksGjGCPoaum90B0GkS3Z1C8F8QjvHHMsYPEYO4Y+vAz9a2gwPI5z0qZLURWka6GoRJHGhtTGTI5PO7PAFLbLNHvNxKHLSHaAMbR2FSBZ4xWbZyf8TjUoCfumNx+K/8A1qANKkwBQMMcU3bzQBmXqhb9Td7vsroFRgSAr553Y9eOamsLWW1knVpS9uSDEG5K+oz3FMC2jszuChUKcAn+KnmkBT1DLWT47kfjyKq6N/o8l5YE/wCpmLp/uP8AMP54qugGtSVAFuqkun281wJ5VZnHTLHA+gqk7CJBaW6tv8lC394jJqfr06Ci4AeBk8e57Vlalcabdx/Z5bsPzzHA29m9iBmmtwKVjLMimySO+ZoR+73kR5Tt1qy1ldyWT2kaRQxsxbLuXIJbPYCqbSAS8sb11jkkvSGjcNiOML14Pr2Jqx/ZkR4kkmk/3pDj9KltDFTTbKI7ltYt394rk/rVjGBjt6VICGqcOnpDqd1eocNcqgcY7qMZpp2AyvEFsxvYW2M8NzEYJFHchg4H4gNVXTrvUVa3tLOF/ISeRUaSMhDFt+XJ7EH1rRaoC/Bq5n1KygaRfMZJBNGvZh0+nSlFjOYdUjBk3Lc+fbMfUqDx+OR+NQ1qA291DU4dO+2GAxRMdrRbcvGp/iP+FQ6Nthv9Qvft4uYJUjWF2YZYjcSBjr1xVaWAn0m71C5uVMzLggiaFjhoj2OOuKkitrpmu4Ir6QRoP3ZGDyR0z7GjRAWrC9Qwx207+XdRqA8chwSe5Geo+lX+nJrN7gREw3KMgKSqRhhkEVS+fTZFAbdauwTDHmInp9RTAqg30S3DgyStb3W8IB9+M9VFTXV1KbdvPglW3mUqNqlnX3IFNLUCrpU0l5bR2bgK1s2JznIPdQPqMfkfSr99Yec4ubc+VeRjCSf3h/dPtQ3ZgS2V39rjcMpSaM7ZYz1U/wCB7VaqWBbqldzXsbhbW0WUdSWk2imtxEMM2rznDW0VvjozsWz9MVMtncyczX8x9VhAQf403boAv9mWecvF5p9ZW3/zq1GkcS7Y0VB6KABSuAhRGkWQjLLwGzS5BpN3AQ4IwcGkJpDGmmkimAhIoyO9IBjIkhXeudjbl9jTXniiYK7hD23cfrVXApX0kBnsniaMyG5UfKRkggg1pAqMcjnge9DYAWHUnGeOe9JsQHIRR74pARTWtvMwaWKN27FhzT1EcK7ECov5CgAkijlXEiK49GGaRIkiHyLtHoDxQAyS3gkbc6An1zg1GLG23q3l7mByCzE4oAsiqeoWK3exwsbSRnKiRcqfrQnYBLKCZZ5riaNIy6qgRDnAGf8AGr1DAqXFr+/juoRiZOGx/Gp6g1Z7c0Af/9kA/+Ex6Gh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRvclRvb2w+V2luZG93cyBQaG90byBFZGl0b3IgMTAuMC4xMDAxMS4xNjM4NDwveG1wOkNyZWF0b3JUb29sPjx4bXA6Q3JlYXRlRGF0ZT4yMDIzLTA1LTEwVDIyOjQ4OjI1LjQwMDwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PC9yZGY6UkRGPjwveDp4bXBtZXRhPg0KICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMAAwICAwICAwMDAwQDAwQFCAUFBAQFCgcHBggMCgwMCwoLCw0OEhANDhEOCwsQFhARExQVFRUMDxcYFhQYEhQVFP/bAEMBAwQEBQQFCQUFCRQNCw0UFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFBQUFP/AABEIADUBGgMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APr3IoyKTdRurnLA4NN2inbqSgBuwUCMCnL83QZ659sUfjmgBpUGozDU1NqBkHlHcUBG7jC+uag2xyKHVtydcqc8Zx1+tTXEgjjcnbgAt17AHOfzrhPg7eNP4FsbeYyNNayT27F+WIEpKE/8BxWqhdXGdk8QUdTTKmk9KibioAZ5Y96AqrSlsCjdUgIYw1Hlil3U/FADfJFJ5OPWpx0FFACbQQMntQVU0u6hWJ9KAG+WtL5amnbqVWBoAj8lfU1H5IDHk1Y3Co2xQBA0aluaTyUqXIpjNigCPyVoKDkU/dSbaAI/JWjyVp9FMCPaBkUvlrTttGRSAYUHSo2iU1NTeKAK/kLTfJFWNtNoA6OiikbtjBH94nA+lWSNZitMaYj0onkSBGkmdYoh1kdtqj6k4rnG8XW95dPbaVbXGszJkM9qpWAH3lYBfyyPc1STA4b47eONS8C6h4J1Wwmd1bUXtJrNf9XOjpzuPqOwr0jwz4w03xZo9vqmnszWsx27XBDxyD70Mgx8jqcg59DXmnxq8NeI/GHw/wBYjn/s3TreC3+2wwgNcT+bFh0KSAoqnqOhyPzrmdI0Cx8WeE9N8YaX4vg0rUdUhW7vbTXJYWtbiQAF0aP5SrZJG4fjmu1UoypeYHt+oeNNHsLj7ML1bu8zj7LZKbibPoVQHb+NQ/8ACQanc5+zeHbpQTw15IsI+hADMPxAPtXLeEfi14VuNJMT3djo95bN5F3Z23zJG443Ky/ejbGQegGOtdFrlk2vWei39hrbabbw3UV+bi3ZRHcRFBhTkcqwI6YPNc7p8oHO+Pb7xr/wjGqT6ZPoemNHbSOBJHLcsxA6BlZMH6iovh3ps2k614ijuHxIz229VJADtbR7iB2y26uo8dMYfC2oj5lLx+VtBPR2VcfX5j+VU7WH7P451qH+Ke2tblQeh2tLGf18v86raIG/5rHrgmmtJRtFRyLtPzcD6iuYoY8hqP7RjinldxwoZh3wCSB+GfzzUTxkE0vUBVvAZFQYJY7VG7BY+gqwlzuAxjFeC/tRXOuaPo/hDU9BlaK/ttbUKik/vsxO20gdQSldPrnxosNL8F6Rq1qn2nVNSgjni03OWRQR5zSEfcEfI57it/YvlUl1A9ZSUsDTGuSCRgUkZCr1yOx9aVkBXj7xP4VjawDWm70JKfWudj8c6Te23iGSzkku20OaW2u444zu82MfMig/eOeOK1tGum1TTLe7ktprN5UDm3mGHQn+Ej1FAF55SqgioxOd+Mjk46VJJHxtOeK5zQNXn1LxB4k0+dlIsbqERKBg+W0COCfxLfgBRbcDokmMh46dqbJcbW205Y1Q8ce1MaEu5JpAR/aDTWmNTeQvvUUkVMCP7QfSlM57GqPiDUI/D+h3+pSQvMlpA07JHyxwBx7devpXK6vqXinR9G/tqOLTtWt0iF1LY2ausxhIGTG5cq+M9hzSsB38bblGeuKdVPTLqO+0y2vU3rBNCswMo2FFKgjdnocGrMMiTRJJG6yI6hldTkMD0IqQH1zPxE8TXHg3wdqus29st1LaIrrEzfey6r0/E10rMFrmPiAqXPh3yJE8yOe6ton4/haZVwfbmtYL3lcDe0u+h1jT7W9tm3W9zCk0TZzuVwCvP0NT5zyOlcL8FZzb+Dv7Fl3CfQb240wq3XZHK3lY+sYXFd104P6VM48sgE4pu2n5FRluTUAdBurC8TXXiBVSDQbG1leT/WXl1PtWEey4yT+NblG0MCxBYAZOMnPPFaLck4mPwbqd86zXlxa3t31M+pM9yq+yRgxqB6A5/GtdPC1zJGFu/EF+YRwLWxSKzhH/AHwm4f8AfR+prflYRBvOZUxzmQhR9c5rn7/4heG9Mk8mbXLSSfOPs1u/2iXPptj3HPscGr5pIB48E6OwIntpLzJBb7VcyzZ9fvMevtTfCfw/0DwVp72Wl2McMDzPPtcbypYkkAnoMnoKani6bUELaZ4d1i7XqJLiAWsePXMhB/Ska58W3iAW9jpGlq3VruaS5b8FUIPyYj3NF5dwPPviDb6rB40/tF9M1Gz8NPs0/VpbJ4i13AeVc8naA2B0B9a63xlBfa54P1HS7Pw/eW7GBRbkSQCOIx8qD+93AfLjGM4q1eeBdR162lt9d8TX0tvOhjks9Nt0tINpOcEEM555zuzVPw78MbC3szba1aNrFxbSMkN1qEzztJGSxjyGJGdp5wAPYV0OacU2BB4+8aaRavosV3qVmqS6lB9tjWYSPEqqz/dXPAdVBz79KyNR8dWreL9FvdO0/V9USa0ubDdDYSRLIzmOWMhpQqnHlSYIOOa9IsNJsNLULZ2dtbADb+4hVDj0yBmm6tYnUFt2yGe3uEuUL8/MBjv6gnP1NYe0j2Gc3JrHiO43CDw5Hatn72oX6If++ED/AM6rNp/jTUAc6ppGkp3+y2T3Ev8A3077Qf8AgNdk0ax5C8DtjiopBuHPP4AVlzeQziz4AmvFJ1PxPrt7IG3KEmjgjDeu2NBx7EmupjiMNvGjO0rKoUyN1YgdT7mrDfdAHApmNwpOTe4Hl/xw8PXetab4UubRGl/s3xHZXcqIhJMal1b/ANDP5CruvfC/Q9N8M+MpdJ0yKK/1WyuVlk5JbKMSBk/KpY7sDHNegbRznkccHpwc04BTww+U5BHqDjIP5VrGo0kkB47Z/GLU1vtIvrmOxXwe8VjbXF2Q5uRcXMMbI/B2hNzlTlevevQbD4h6Dfard6WNQhh1G31CTTjaSvtl86MAkYx3zxXEp8Co77R7nSLu+u7XTHvZFeCLBjuLVJfNt1J6hk+VcjHAxjvW5H8JbXT9L1FbK8kk1eS8j1BNSv0SaUSRlQqsyqpK7VAPc881tL2b1A1IdZtHsfFsWjWMUl3p1zOJrdhgTXRiEm49zuzzzWdpfxBudU0vwNrv2ZU0jxBBHHNuJ32k80YeIk55QklT3yOtQfD/AEvUNB8aeKrLVbyPULq9e31ZmhhMcQ3homVASTtAjTqSeTz6bi/D3S38Cw+EpGnOlwwrDGyuBIu196sGxwQc4471jZJgJqXxK0GxNzELz7ReQ8fZ4gWaV9wXYh6Fsk5GeK474W+JpPEnxU8cs+l32kSiCxElrqCiN1ceZFx1BB2g59K6LxZ8K9O1zRdIsNNmfRLvQ5Un025hTzAjAgkMhPzhsc55PqKsaH4V1rR5b66uL6wm1a/KC61BbWRWdVGFRVLnYBzgc4q04coD7P4m6Le61FpsbyZnlltra6YAW800Zw0e8E7SSCASMEimXPxCTT9eu9JvtMv1nUlrM2sDTC6i4w4wPl6jI5rF0z4YnwkyQabZ2msaUs5uYor1zDcQzFizlZQMFdxJwVJ966qy028vvEEOr30cNqbaCSC3t7Ztx+dlZyzAAdVHTHU0nyJAL4V8UL4nh1CQWU9kbO7ezMU/EhdNpYFcccHitrdn/wDVXOXnhW4t9Xl1TQdS/s68uCGuoZozLbXR4+ZgCGDYAG4Nj2q9aXWqptS7sLfHQyW05Kj8GH9T9aylZvQDUdVaMqyq6EYKsMgg9QfUH3rjJNJvfBheTToW1Pw8wMj6avM9qejGAcbl7mPj2NdPqH2/ar2RhAHBjnQ5b6MGwPyNUZNT1KOPnRJHk9YLmNs/mQfz6e1ICh4g8a6ZpngO88RRj+1NLhtWnaK2yzSIuMqB647dsVc0vXrb+1m0SG3aDyLGO7gbP7t4Sdvyn/Yxk+1ZNh4budQufEK31ium6dq0CxvaLIrM7ncryEDhSd2cDvjrVm++H9hqlposdzcXJn0uAW6XML7Glj2BWV+OVbGSOOTVe6AvizxkNN8M6jqWkm2vpLORVlDMSkYyNxOOuFyfyqPWNUTWLHQbeHa0mpSQXKheQsaFZi59ugzXOeKNHabxhHptkF0q2uLdI5Q18sMF6gUr5ZjaMlvlO04YEgDnvV34V6B/Yba5ZNKLyXTbt7KCbBHl25RHWFQSSApYpzk4UDOck62UY3QCa0uofD7xPqHiRLf7b4ev/KbUIoVPn2siDYJl5wy4+8MZ967y2uob23jntpUnglQSRSIciSMjIdfUYp0kaybwyhgw2tu5yOn8q4m3tZPhvqR8lWfwldzBTEOf7Lmbow7+S3dex6EDis3JT3A7jcDz0/WmHqakPygjGCvBGc1FWQHRbaxvE3hOx8WQxQag9yYIwR5cM7Rhs+u05ooq72JOZtvgb4Vs5xcJFePFGwAtJpxLCT6kOpY/99V21rp9poawLZWsVqkmFC26iPHp07UUU3JvcouSxkSHLbm3BdzDceT75qqtwJNmFI3Dd1yR+OKKKkBzTYjDBf8AWFh7jA9etPl4B2nAIyOckfj60UUrAVlud6lwuFBb5c+gz1p2/cVGMAmiipAimby9oPO59np+NRuw3AY4Lhevuc0UUwIFkLTLGR/yyEpP/AkGP/HjSw5uIwVOzLiMZGcEruz/AEx+tFFICKGb7REsgG0MN23Oeynr/wACoRzJsHQseD6cUUUwJNrSIzAhWC7gducVwVx8TJ9N1F7SexjnkRiolhkMQIHHK4b+dFFUBF4T8Rf8JF8Rr66+z/Z/K0eOHaZC+Sbhm3ZwPp0rvVvDJqQswgBMJlEmf9lDjH/Aj3oooAbe3n2O7soCvmfaZdgbI+T5WPoc9Para5kiMoO0r8p75oooAq318LK0M5j3lOQA2Oqk+/cUalcGyhmcgusCltucZwWHv/dHrRRQA2xvvtlms+zYGUHZnoS2O2PyqwyFV3Fsn2+Xv7UUUrARzP5EsiNlgrqG2nbuyetRrcF5VjI6jOcniiikSOjXco/OndOBRRQUUNY0Wy16za0v7dbm3zkK5OVPqrZyp+hFO0XRrPQLIW1jF5Ue8yMSxZmY45ZiSSeByaKKL9AL1RXVtFe28kE8ayRSIY2VuhUnOPzoooAZa24tLeOFWZ1jUKGc5JwMZPvTqKKQH//Z)

Signature and Name of the Mentor

Mr. Manjinder Singh(29540)

Designation: Assistant Professor

School of Computer Science and Engineering, Lovely Professional University,

Phagwara, Punjab.

Date: 10/05/23



ACKNOWLEDGEMENT

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this report. Special thanks are due to my supervisor Mr. Manjinder Singh whose help, stimulating suggestions and encouragement helped me in all time of fabrication process and in writing this report. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

I have achieved a good amount of knowledge through the research & the help that I got from my supervisor, friends and family.

Apart from this, I would like to express special thanks to my parents who have supported me and helped me out in my project despite their busy schedules.

Last but not the least, I would also like to thank my classmates who had my back all the time in accordance with the project.

ACCEPTANCE LETTER

Graphical user interface, text, application, email

Description automatically generated

# TABLE OF CONTENTS

Inner first page I

PAC form II

Declaration III

Certificate IV

Acknowledgement V

[Table of Contents VI](#_bookmark0)

1. Introduction 1
2. Profile of the Problem. Rationale/Scope of the study (Problem Statement) 3
3. Existing System 4
   1. Introduction 4
   2. Existing Software 5
   3. DFD for present system 6
   4. What’s new in the system to be developed. 7
4. Problem Analysis 8
   1. Product definition 8
   2. Feasibility Analysis 9
   3. Project Plan 11
5. Software Requirement Analysis 12
   1. Introduction 12
   2. General Description 13
   3. Specific Requirements 14
6. Design 18
   1. System Design 18
   2. Design Notation 19
   3. Detailed Design 20
   4. Flowcharts 21
   5. Pseudo code 23
7. Testing 25
   1. Functional testing 25
   2. Structural testing 27
   3. Levels of testing 28
   4. Testing the project 30
8. Implementation
   1. Implementation of the project 31
   2. Conversion Plan 33
   3. Post-Implementation and Software Maintenance 34
9. Project Legacy 36
   1. Current Status of the project 36
   2. Remaining Areas of concern 37
   3. Technical and Managerial lessons learnt. 39
10. User Manual 40
11. System Snapshots 41
12. Bibliography 51



Introduction

Weather prediction is the process of forecasting weather condition at a particular location and time in the future. The goal of weather prediction is to provide accurate and timely information about weather conditions to individuals, organizations, and governments so they can make informed decisions about their activities and prepare for potential hazards. Weather prediction is based on a combination of observations, physical laws, and mathematical models.

There are two main types of weather forecasting: numerical weather prediction (NWP) and statistical weather prediction (SWP). NWP uses mathematical models to simulate the atmosphere's physical processes, such as convection, radiation, and atmospheric pressure changes. SWP uses historical data to capture patterns and relationships between weather variables and uses statistical techniques to generate predictions. Weather forecasting has many practical applications, including aviation, agriculture, transportation, and energy management. Accurate weather predictions can help farmers plan their planting and harvesting schedules, airlines plan their flight routes and schedules, and energy companies manage their power generation and distribution. Additionally, weather predictions are crucial for emergency management, allowing authorities to prepare for and respond to natural disasters such as hurricanes, tornadoes, and floods. Weather predection remains a complex and challenging task. Accurately forecasting the weather requires a deep understanding of atmospheric processes, access to huge amounts of data, and sophisticated mathematical models. are helping to improve the accuracy and speed of weather predictions, but there is still much work to be done to improve the reliability of these predictions, particularly for extreme weather events.

Weather prediction is involves analyzing large amounts of data from various sources such as satellites, weather stations, and radars. Machine learning has emerged as a powerful tool for weather prediction. There are also many challenges associated with using machine learning for weather prediction. One major challenge is the availability and quality of data. Weather data can be sparse, noisy, and inconsistent, which can make it difficult for machine learning algorithms to learn and make accurate predictions.

Overall, machine learning has the potential to revolutionize weather prediction by improving accuracy and speed, enabling more informed decision-making in critical situations such as severe weather events. However, there is still much work to be done to overcome the challenges associated with using machine learning for weather prediction, and researchers are actively working to develop new and more advanced techniques to improve the accuracy and reliability of weather predictions.

It involves developing mathematical models and algorithms that can automatically identify patterns. We used seven models for predicting the next day’s weather that rainfall happens or not by taking extensive historical weather data from Kaggle. We take this data and use it to develop and find a simple learning model that can predict the weather for the next few days. These models can work with low-cost, low-cost equipment while providing fast and reliable predictions that we can use in our daily lives. Short-term weather forecasting using machine learning, capable of running on low-cost machines, is an important part of this document. Will use an extensive historical data from Kaggle. Comprehensive evaluation of the proposed method and comparison of various machine learning models for future climate prediction. For result evaluation and comparison of different model’s roc curve and auc score is used.



Profile of the problem

The problem that this project aims to find is effective weather model for weather prediction among all forecasting algorithms of machine learning. Looking at today's results, it has been shown that tools based on machine learning (ML) techniques are more effective in predicting weather conditions than traditional models based on weather. This project includes algorithms like- catboost, xgboost, random forest, knn, svm, naïve bayes, support vector classifier etc. We use distribution indicators - the roc curve and the auc score - to compare different models and choose the more precise model for climate change.



Existing System



Introduction

Weather prediction systems use a combination of advanced technology, including satellite imagery, radar data, and numerical models, to forecast weather conditions accurately.

In addition to numerical models, meteorologists use a range of other tools and techniques to predict the weather. These include:

Doppler radar: This type of radar uses a special technique to detect the movement of precipitation and estimate the speed and direction of wind.

Satellite imagery: Satellites provide meteorologists with detailed images of cloud cover, temperature, and other atmospheric conditions.

Weather balloons: These balloons are equipped with sensors that measure atmospheric pressure, temperature, and humidity at different altitudes.

Surface observations: Weather stations located on the ground provide data on temperature, wind speed, precipitation, and other factors.

Historical data: Weather patterns tend to repeat themselves over time, so meteorologists often analyze historical weather data to identify patterns and make predictions.

Overall, weather prediction systems rely on a combination of technology, observation, and historical data to provide accurate weather forecasts.



Existing Software

There are several weather prediction software available in the market. Some of the most commonly used weather prediction software are:

NOAA's National Centers for Environmental Prediction (NCEP): It is a suite of numerical weather prediction models that provide global and regional weather forecasts. NCEP models use a range of data sources, including satellite data, radar data, and weather station observations.

ECMWF is a European-based organization that provides numerical weather prediction models for global weather forecasting. ECMWF models are widely regarded as among the most accurate in the world.

Weather Research and Forecasting (WRF): WRF is a mesoscale numerical weather prediction system that provides high-resolution weather forecasts for regional areas. WRF models are commonly used by meteorologists for short-term weather forecasting.

The Weather Company's WSI Fusion: WSI Fusion is a weather prediction software that uses a combination of machine learning, artificial intelligence, and weather modeling to provide accurate weather forecasts.

AccuWeather's AccuWeather Enterprise Solutions: AccuWeather Enterprise Solutions is a suite of weather prediction software that provides weather forecasts for businesses and organizations. AccuWeather's software uses a range of data sources, including satellite data, radar data, and weather station observations, to provide accurate weather forecasts.

These are just a few examples of the many weather prediction software available in the market. Each software has its unique features and capabilities, and meteorologists use a combination of these tools to provide accurate weather forecasts.



DFD for present system

Diagram

Description automatically generated

Figure 01: Data flow diagram



What's New in the System to be Developed.

Our new text to image converter website represents a significant improvement over the present system in several key areas. These improvements include:

Use of the seven Model: The present system uses a conventional machine learning model to make prediction. In contrast, our new system uses seven model to predict result by feeding numerical data in system, which has been trained on a much larger dataset and is capable of generating more accuracy with have greater diversity of weather features.

Improved User Interface: The user interface of the present system is relatively basic and lacks advanced features such as real-time feedback and features editing tools. Our new system features a modern and intuitive user interface that is designed to be UI friendly and accessible to a wider audience.

Real-Time Functionality: The present system predect rainfall from a batch process, which can take several minutes to complete. Our new system features real-time functionality, which allows users to see the predict as they type their description of weather features.

Broad Features: The present system has a limited features to predict, which can restrict to access proper condition in weathers and ultimately less accuracy. Our new system has a much broader features , which allows to make prediction more accurate with more diverse and realistic result.

Overall, our new prediction website represents a significant improvement over the present system. It utilizes the latest machine learning techniques and technologies to generate predictions in real-time and features a user-friendly and intuitive interface that is accessible to a wider audience. In the following sections, we will provide a detailed analysis of the existing system, problem analysis, and projectplan, as well as a comprehensive evaluation of our new system.



Problem Analysis



Product Definition

The product we are developing weather prediction website that uses the seven model to generate result from numerical data in system in real-time. The website will have a user-friendly and intuitive interface that allows users to easily input their numerica descriptions and view the result.

The primary target audience for our product includes individuals and businesses that require high accuracy to predict rainfall condition for various purposes such as agriculture and farming, travel planning, outdoor activities etc. The product can also be used by individuals who simply want to check weather condition for personal use.

Our product will differentiate itself from other weather predection by its real-time functionality, high accuracy, and broad features. The real-time functionality will allow users to see the predict as they type their description of weather features, which can be especially helpful in creating closely matches weather condition in suurounding . The use of the seven model will ensure that the prediction considered and add most of aspect of diifrent model of machine learning in website. The broad feature of weather will provide greater diversity to improve accuracy, allowing users to set awide range features of that suit their needs.

In summary, our product is a weather predection website that uses the seven modelto generate result in real-time. It will have a user-friendly and intuitive interface and will target individuals and businesses who require high-quality result forvarious purposes. Our product will differentiate itself from other rainfall prediction by its real-time functionality, high accuracy, and broad features. In the following sections, we will provide a detailed analysis of thefeasibility of our product and the project plan.



Feasibility analysis

The proposed project uses HTML, CSS, Bootstrap for building the client-side user interface, uses Flask for building server-side logic and IDE use for this projext is Jupyter notebook, Pycharm.

The app aims to predict rainfall, allowing users to view their results. The project requires skills in web development, machine learning and problem-solving. It also involves teamwork, effective communication, and collaboration. The project's novelty lies in combining several cutting-edge technologies to create a unique application that can be useful in fields like art, design, and related fields.

The project requires resources such as a computer, code editor or IDE, HTML, CSS, Bootstrap, Flask and seven machine learning models. Once these resources are available, the project can be carried out by setting up the development environment, designing the overall architecture and layout of the application, setting up the server- side logic, building the client-side user interface, deploying the application on a hosting platform, testing and debugging the application, and maintaining and updating the apppost-deployment.

1. Economic Feasibility: The project refers the cost- effectiveness of implementing the project. The project is expected to be economically feasible as it has a potential market and can generate revenue through paid subscriptions, advertisements, and sponsored content. The cost of developing the project and maintaining it may depend on various factors such as the cost of the server, the cost of the domain name, and the cost of hiring developers to maintain the project. However, these costs can be minimized by utilizing cloud-based platforms like Cloudinary for storage and retrieval of output result. Moreover, the potential revenue from the project is expected to outweigh the cost of implementing it, making it economically feasible.
2. Technical Feasibility: The technical feasibility of a project refers to its ability to utilize the available technology to implement the project effectively. The project is expected to be technically feasible as it is based on the flask framework and use pre-existed machine learning model which work on any devices.
3. Operational Feasibility: The operational feasibility of a project refers to its ability to be integrated and sustained within the existing organizational structure.

The project is expected to be operationally feasible as it is a web-based application. However, it is essential to ensure that the project is user-friendly, easy to use, and can provide an intuitive interface for users to generate result .

Overall, the project is expected to be economically, technically, and operationally feasible, provided that it is developed and implemented effectively, considering all the necessary factors.



Project plan

Define the scope of the project: Before beginning the project, it's essential to define the scope of the project. This step will help to ensure that everyone involved in the project understands the objectives and goals.

Create a timeline: Develop a timeline for the project that includes all the necessary steps, deadlines, and milestones. This will help to meet deadline.

Identify resources: This step will help to ensure that the project has the necessary resources to be completed successfully.

Assign roles and responsibilities: Team members toensure that everyone knows what they're responsible for and can be held accountable.

Develop a communication plan: Create a communication plan to ensure that team members know update, deadline and changes in the project's scope.

Establish a monitoring and evaluation system: Develop a monitoring and evaluation to track the progress of the project and it’s meet the objectives and goals set out at the beginning of the project.

Identify potential risks and develop a risk management plan: Identify potential risks that could arise during the project and develop a risk management plan to mitigate them.

Create a budget: Develop a budget that includes all the costs associated with the project, such as personnel, equipment, and supplies.

Obtain necessary approvals: Obtain necessary approvals from stakeholders, such as management or clients, before beginning the project.

Begin the project: Once all the necessary steps have been taken, begin the project according to the established timeline and monitor progress regularly to ensure that the project is on track.



Software requirement analysis



Introduction

It is an essential process in software development that involves the identification of the goals and objectives of a project, and the determination of the software requirements needed to meet those objectives. In this proposed project, a seven machine learning models is used to get best result for users. The application will be designed to provide students with an in- depth understanding of the frontend as well backend with several ML models, to create advanced features within web applications. The project will require the useof skills in web development, machine learning to build a production- ready web application. This proposed project is unique in that it combines several cutting-edge technologies to create a functional and visually appealing web application, a unique feature that can be useful in many fields like agriculture and tourism and other business related fields. We can carry out the proposed work in our own device by setting up the necessary development environment, designing the overall architecture and layout of the application, and integrating the necessary technologies to achieve the project objectives.



General description

The proposed weather prediction web application requires a set of functional, non- functional, and technical requirements to be met to provide users with a user- friendly and efficient way of predicting and viewing content. The application should be developed with security, performance, and compatibility in mind, while also taking into consideration time, budget, and resource constraints. Future enhancements could include the integration of third-party APIs to provide stream of data.



Specific requirements



Functional Requirements

* Input Text Field

Input: The user should be able to input text in a designated field. Output: N/A

Processing: The application should provide a designated field where the user can input the text to be used for prediction.

* Visual Design

Input: The application's user interface.

Output: A visually appealing and user-friendly design.

Processing: The application should have a visually appealing and user-friendly design created using CSS Bootstarp.

* Testing

Input: The application code and test cases.

Output: Verification that the application isfree of bugs.

Processing: The application should be thoroughly tested to ensure that it is free of bugs.

* Maintenance

Input: Updates and maintenance tasks. Output: A functioning and secure application.

Processing: The application should be maintained and updated regularly to ensure its continued functionality and security.

* Documentation

Input: Instructions and troubleshooting information.

Output: Comprehensive documentation for installation, use, and troubleshooting.

Processing: The application should be thoroughly documented, including instructions for installation, use, and troubleshooting.



Non-functional requirements

Performance: The application must generate result quickly and efficiently. It should be able to handle a large number of requests without any significant decrease in performance. The response time for image generation should be no more than 5 seconds.

Scalability: Handle a large number of users and requests. It should be scalable to accommodate an increase in users without any significant decrease in performance.

Security: The application should be designed with security in mind. It should implement secure communication protocols (such as HTTPS), and enforcesecure authentication and authorization mechanisms.

Usability: It should have a user- friendly interface and provide clear instructions on how to use the image generation feature.

Compatibility: It should be tested to ensure that it works correctly on popular browsers such as Google Chrome, Mozilla Firefox, and Safari.

Maintainability: It should be designed with modularity and clean code in mind, making it easy for developers to addnew features or fix any issues that may arise. It should also have documentation to aid in its maintenance.



Design



System Design

Designing an website involves several components and subsystems that work together to produce the final output. Here is a high-level system design for an website.

* User Interface: The user interface (UI) is the front-end component of the system that users interact with. It consists of a web page that allows users to enter text and submit it to the system. The UI also displays the generated image as output.
* Input Processing: The input processing component receives the user's text input and performs various pre-processing tasks such as cleaning and tokenization. The input processing also extracts relevant features from the text, such as the sentiment, keywords, and entities.
* Generation Model: The generation model is the core component of the system. It is responsible for generating the result based on the input textand the extracted features.

In summary, website involves several components that work together toproduce the final output. These components include the user interface, input processing, result generation model, storage, user management, and deployment.



Design Notation

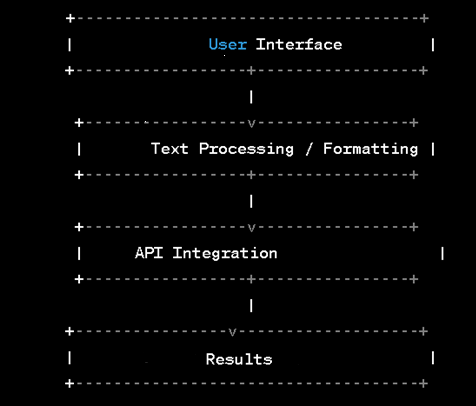


Figure 02: Design Notation

In this design notation, the user interface component provides the interface for the user to enter the text input. The text processing and formatting component pre-processes and formats the text input to prepare it for the generation service.

The generation service component is responsible for generating an result from the input text. This could be done using machine learning or other techniques. The output processing component formats and post-processes the generated image, which is then presented to the user as the final result.

The generated result component represents the final output or results of the system.

The purpose of this design notation is to illustrate the main components and flow of data in the seven-model website. By breaking down the system into smaller components and their interactions, it becomes easier to understand and modify the system as needed.



Detailed Design

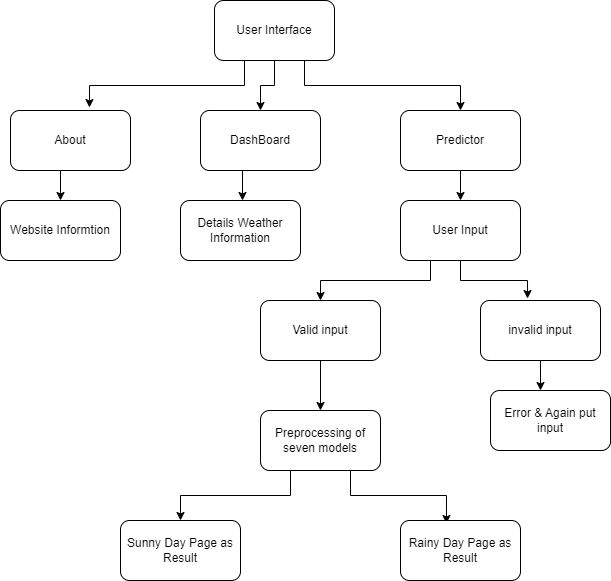


Figure 03: Detailed Design



Flowchart

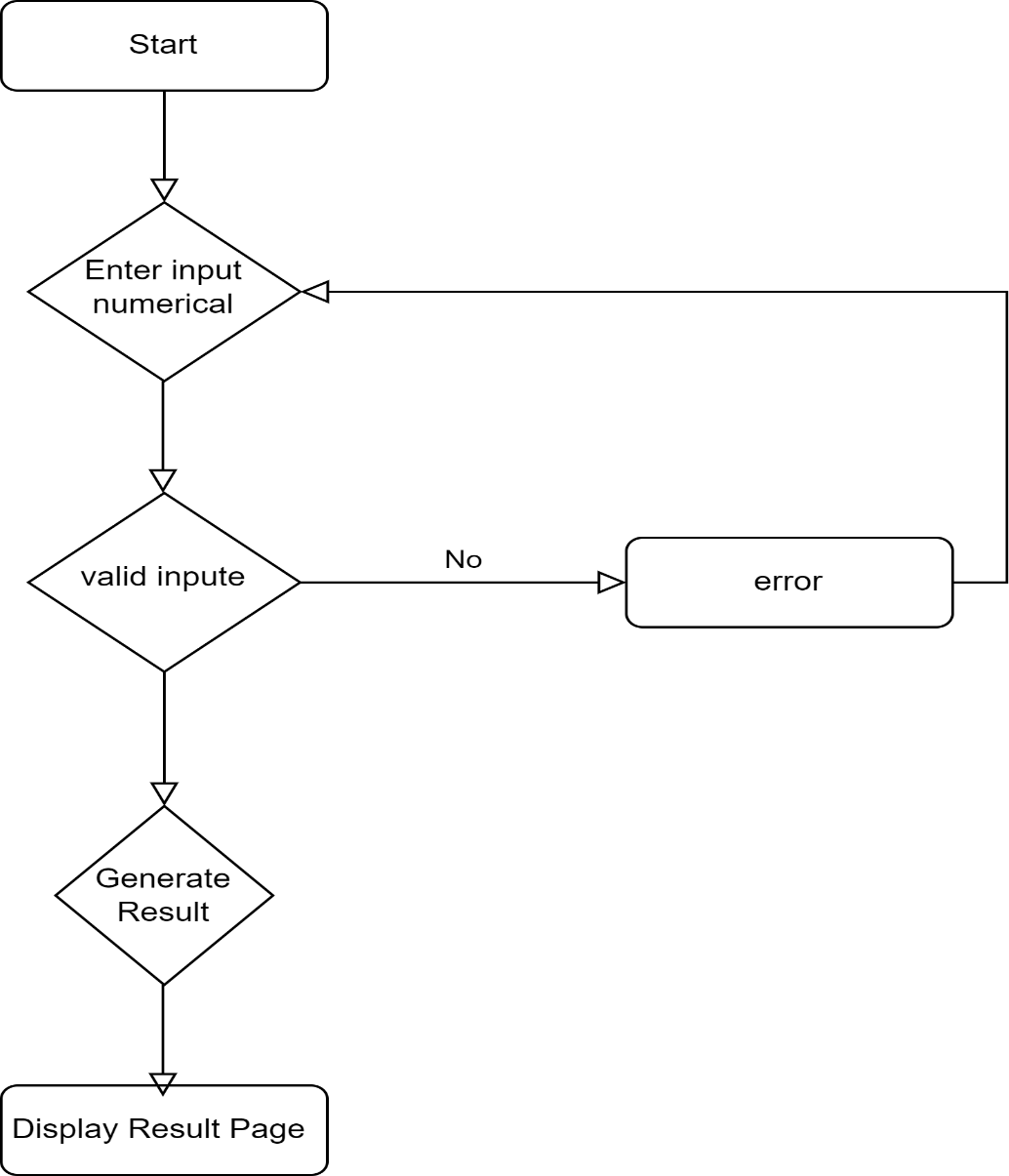


Figure 04 : Design Flow Chart

1. The flowchart begins with the user inputting values into the input box
2. The input processing component of seven model then validates the input values to ensure that it is valid and safe for processing. This step is important for preventing errors and ensuring that correct input pass.
3. The input processing component of seven model then formats the input values to make it suitable for predection.
4. The result generate from the analysing values using seven ML model. This step is the core functionality of thewebsite and is responsible for generating the result that the user will see.
5. Finally, the output component displays result page.



Pseudo Code



// Set up variables for text input and API key let textInput = "example text";

let apiKey = "your\_api\_key";

// Call ML model to generate

let generatedModel = API(textInput, apiKey);

// generated result

displayResult(generatedresult);

Figure 05: Pseudo Code

This pseudocode shows the basic steps involved in the prediction process, including calling the models to generate the result.



Testing



Functional testing

Predection Testing

* Enter a numerical values in text fields from test case.
* Verify that the generated result with pre-existing result
* Repeat with different numerical values to ensure accurate result is produce.

Seven Models testing

* Verify that the system can successfully connect to the ML models.
* Verify that the system can handle errors and exceptions when the models is down or not responding.



UI Testing:

* Test that the text input field,submit button, navigation bar are functioning properly.
* Test that the generated image is displayed correctly on the webpage.
* Verify that the webpage is responsive on different devices and screen sizes.



User Testing:

* Conduct user testing to ensure that the application is user-friendly and easy to use.
* Collect feedback from users on the design, functionality, and user experience of the web application.
* Use user feedback to improve the application and address any issues or concerns.



Structural testing



Unit Testing:

Testing each module/component of the application independently to ensure it works as expected. For example, testing the result module, the API integration with seven model.

Integration Testing:

Testing how different modules/components of the application work together. For example, testing how the result module interacts with the API integration with seven model.

System Testing:

Testing the entire system as a whole to ensure it meets the requirements and specifications. For example, testing the entire result process, including the API integration, and post creation functionality, to ensure the generated result are accurately displayed on the community showcase page.

Acceptance Testing:

Testing to ensure that the application meets the user's requirements and specifications. For example, testing the surprise me button feature to ensure that it generates result.

Performance Testing:

Testing the application's performance under different conditions, such as high traffic, to ensure it can handle user requests and perform efficiently.

Security Testing:

Testing the application for any security vulnerabilities, such as SQL injection or cross- site scripting (XSS) attacks, to ensure user data is protected.

Usability Testing:

Testing the application for ease of use and user-friendliness, such as ensuring that the text-to-image conversion process is intuitive and easy to understand for users.



Levels of testing



Unit testing

Testing individual units or components of the software to ensure they are working as expected. For example, unit tests can be written to test the functionality of the backend API that communicates with the seven model or the frontend code that handles user input and output.



Integration testing

Integration testing verifies the proper functioning of different modules or components of the software when they are combined. For example, integration testing can be performed to ensure that the API is correctly communicating with the seven model and returning the expected response.



System testing

This level of testing evaluates the complete software system as a whole. It tests the software in a real-world environment to check whether it meets the specified requirements. System testing for your project can include testing the entire web application from the user's perspective, including the UI, functionality, and performance.



Acceptance testing

Acceptance testing is performed to ensure that the software meets the user's requirements and is ready for deployment. It is usually conducted by end-users or stakeholders to determine whether the software meets their expectations.



Regression testing

Regression testing is performed to ensure that the new changes to the software have not affected the existing functionality of the software. It can be conducted after the integration of new features or updates to the software.



Performance testing

Performance testing is performed to evaluate the performance of the software under various conditions, such as high user traffic or load. It ensures that the software can handle the expected workload without crashing or slowing down.



Security testing

Security testing is conducted to identify and fix vulnerabilities and ensure the safety and security of the software. It can include testing for vulnerabilities in the API, web application, and database.



Testing of the project

Testing is an essential part of software development and ensures that the application performs as expected. In the case of our web application that converts text to image using the HTML, CSS and Flask and seven machine learning model, testing is crucial to ensure thatthe application is functioning correctly and providing the desired output to the user.

Unit testing involves testing each component of the application in isolation to ensure that it performs as expected. This type of testing is typically performed by developers during the development phase.

Integration testing make sure that different component interact seamlessly. This type of testing is typically performed during the integration phase.

In System testing is tested as a whole to meets the designated requirements and performs as expected. This type of testing is typically performed by testers during the testing phase.

Acceptance testing requires testing the application from the user's perspective to ensure that it meets their requirements and expectations. This type of testing is typically performed by end-users or stakeholders during the acceptance testing phase.

Security testing confirms that the application is secure and protected against potential threats. Usability testing ensures that the application is simple, usable and appropriate.

By conducting thorough testing at each level, we can ensure that our web application model functions correctly and provides a positive user experience.



Implementation



Setting up the Development Environment

The first step in implementing the project is setting up the development environment. This involves installing the necessary software tools, libraries, and dependencies to develop the website. Some of the tools required for the project may include text editors, programming languages, web frameworks like next js, node js, and database management systems.



Front End Development

The Front End is responsible for the visual presentation of the website and includes the design, layout, and user interface. This mayinvolve using HTML, CSS, and JavaScript to create the web pages, and integrating any necessary libraries or frameworks like react js to enhance the user experience.



Back End Development

After the Front-End component is completed, work can begin on the Back End component of the website. The Back End is responsible for handling the business logic, data processing, and database management for the website. This may involve using a server-side programming language like node js to build the Back End components.

Integrate Seven Models

Once the Back End component is completed, the next step is to integrate the Seven machine learning models for predection.



Testing and Deployment

After the coding, testing is must to assure and check that it’s work what is intended to design to do. This may involve using automated testing tools or manually testing the system to identify any issues or bugs. Once the system is deemed to be stable and functional, it can be deployed to a production environment where it can be accessedby users.

Overall, the implementation phase of the weather predection website involves a series of well-defined steps, including setting up the development environment, front end and back-end development, integration of the seven models, testing and deployment, and user training and documentation. A well-executed implementation can help ensure that the final product meets user needs and is reliable and efficient.



Conversion Plan:

* Define the input text field and create the corresponding user interface element.
* Integrate the seven model with the application.
* Design the user interface to ensure it is visually appealing and user-friendly.
* Set up MongoDB to manage user data and the result generated by user
* Deploy the application on a hosting platform such as Heroku or AWS.
* Test the application thoroughly to ensure it meets functional requirements and is free of bugs.
* Maintain and update the application regularly to ensure its continued functionality and security.
* Document the application thoroughly, including instructions for installation, use, and troubleshooting.



Post-Implementation and software maintenance

Maintenance for the weather predection website involves a series of activities and processes aimed at ensuring that the system continues to function efficiently and effectively over time. Here are some key steps that may be involved in post- implementation and software maintenance:



Bug Fixes and Updates

After the initial implementation, it's likely that bugs and issues may arise. To maintain the system, it's important to address these issues as soon as possible by identifying the root cause and implementing fixes or updates.



Performance Optimization

As the system is used over time, it's important to monitor its performance and identify any areas where improvements can be made. This may involve optimizing the system's code, improving database performance, or optimizing hardware resources.



Security Updates

To protect against potential security threats, it's important to regularly update the system's security features, such as firewalls, encryption, and access controls. This may also involve monitoring the system for any vulnerabilities and implementing security patches as needed.



User Support and Training

To ensure that end-users continue to use the system effectively, it's important to provide ongoing user support and training. This may involve providing access to user manuals, troubleshooting guides, and support resources, as well as conducting training sessions or workshops to address any new features or updates.



System Backups and Disaster Recovery

To protect against potential data loss or system failure, it's important back up data and implement disaster recovery plans. This may involve establishing off-site backups or implementing redundant systems to minimize downtimeand data loss in the event of a disaster.

**System Monitoring and Maintenance**

To ensure that the system is running smoothly, it's important to regularly monitor its performance and identify any potential issues or bottlenecks. This may involve implementing monitoring tools, establishing maintenance schedules, and conducting regular system audits.

Overall, post-implementation and software maintenance are critical components of ensuring the ongoing success of the text to image generation website. By implementing a robust maintenanceplan.



Project Legacy



Status of the project

The project has been successfully completed and is currently live on the web. The application is being used by users to predict rainfall. The integration of the seven machine learning models provided a smooth user experience, with fast image generation and easy sharing of posts.

While the project has been completed, there are still some areas of concern that need to be addressed. One area of concern is the potential for abuse of the app feature to portray 100% accuracy by third party.



Remaining areas of concern



Authentication System

Currently, there is no user authentication system implemented on the website. This could lead to potential security issues and misuse of the application. Adding an authentication system would enhance security and prevent unauthorized access to the website.



Download and Charging Fee

The current version of the application does not allow users to download generated result. Adding a feature to download result and charging a fee could be a potential source of revenue.



Marketplace

In the future, the application could be expanded to include a marketplace where users can not only predict, but also sell their data to big corporate houses.



Security Enhancement

While the current version of the application uses secure protocols, further security enhancements can be made to prevent potential attacks from potential hackers and other security threats.

Multilingual Support

Adding the option for users to write prompts in other languages using third-party APIs could make the application more accessible to a wider audience.



Technical and managerial lessons learnt.

In terms of technical lessons learnt, the project team gained experience in working with seven models catboost, random forest, logistic regression, extreme gradient boosting , svm, k-nearest neighbors, naïve bayes, and integrating it with the front-end as HTML, CSS, Bootstrap and. back-end as Flask.

From a managerial perspective, the project team learned the importance of effective communication and collaboration between team members, especially when working remotely. Regular meetings and updates ensured that all team members were on the same page and working towards the same goal.

In conclusion, the project has been a success and has provided the team with valuable technical and managerial experience. The project team will continue to monitor the application for any issues and make necessary improvements to ensure its continued success.



User Manual A complete document (Help Guide) of the software



developed.

* Welcome to our weather prediction web application! Below are the stepsto use our software:
* First is our landing page as Home from there we explore different options.
* Navigate to the homepage and select predection.
* Click on "Prediction" to go for prediction
* In the " Prediction " page, you will see a text boxs where you can filling numerical values and click submit button,
* Go to navigate bar and click dashboard to check details reports



Source Code (wherever applicable) or System Snapshots

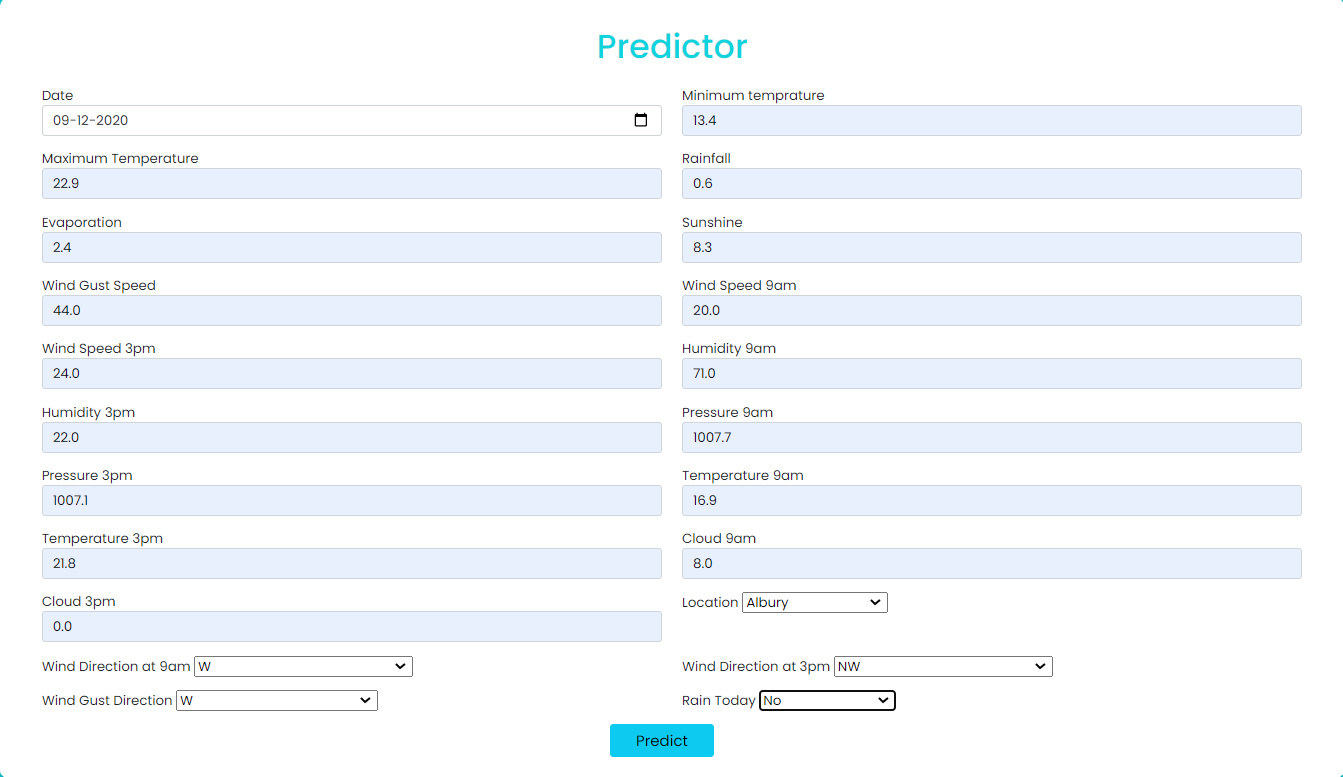


Figure 06: Features Adding.

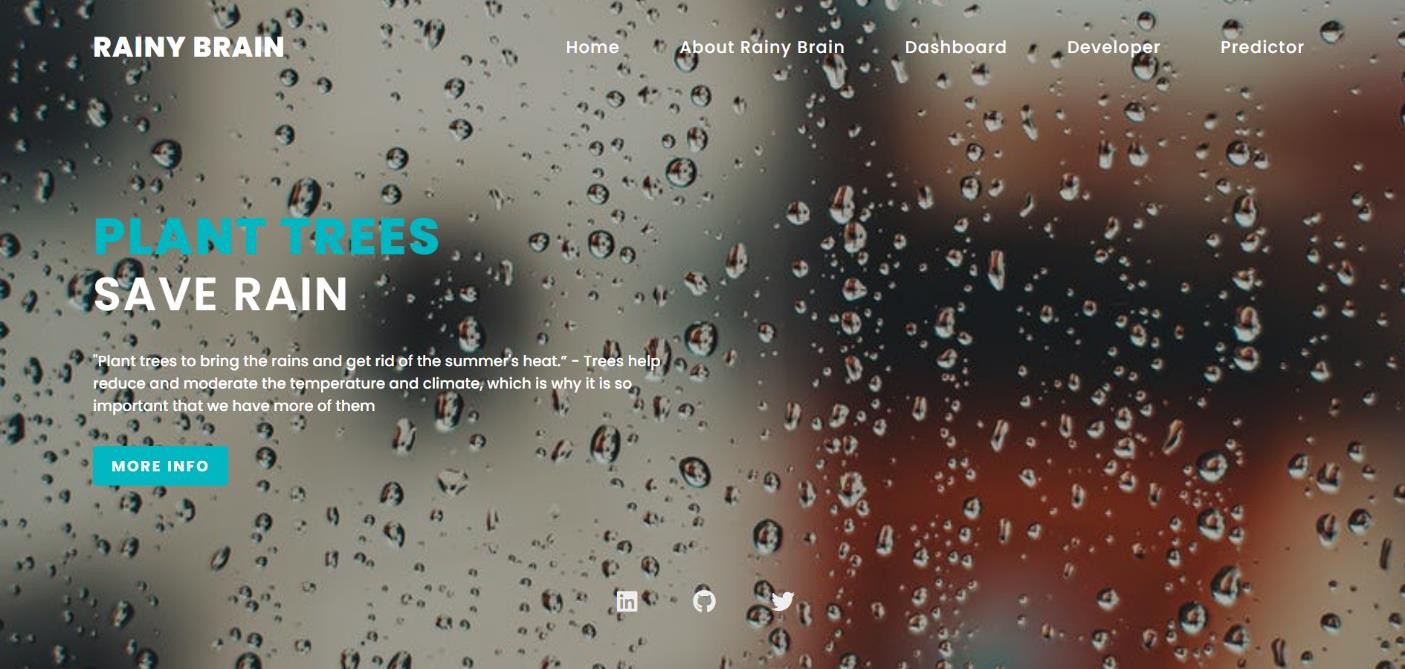


Figure 07: Landing Page.

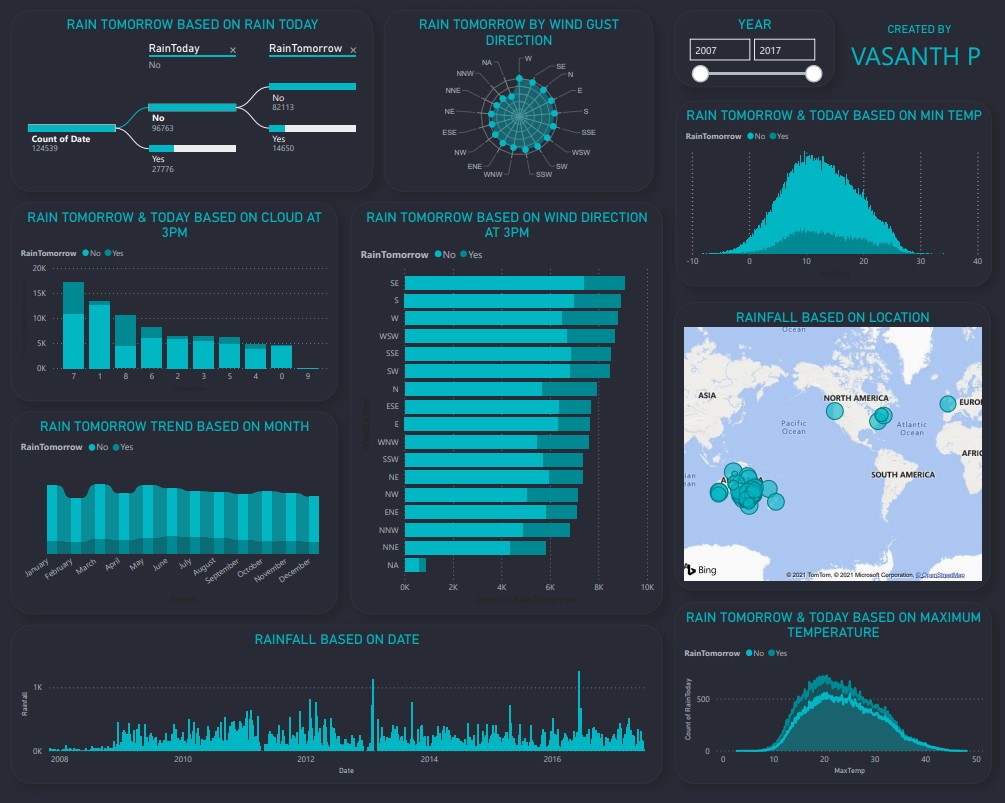


Figure 08: DashBoard

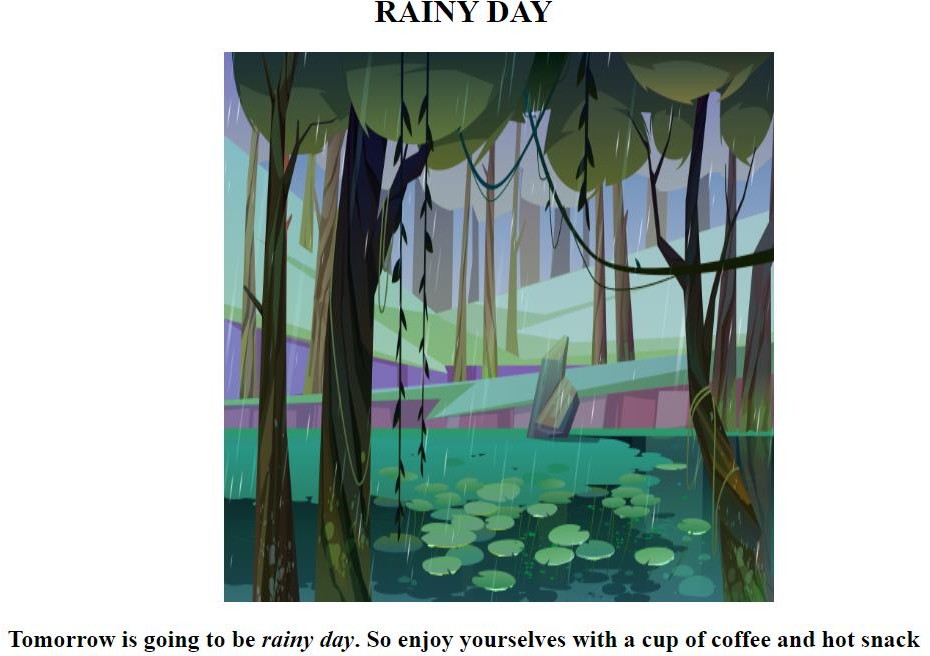


Figure 09: Result Page as Rainy Day

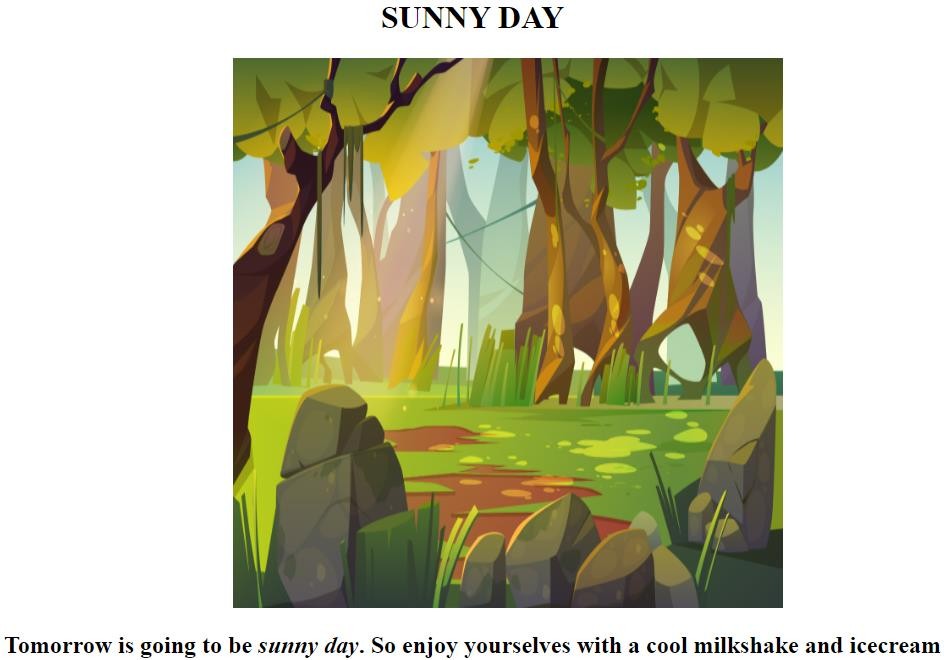
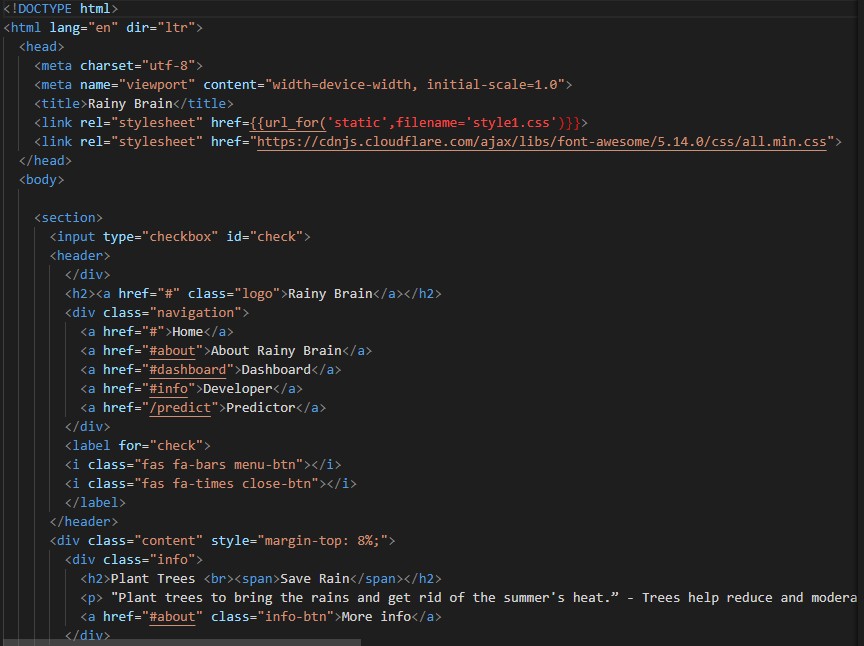


Figure 10: Result Page as Sunny Day



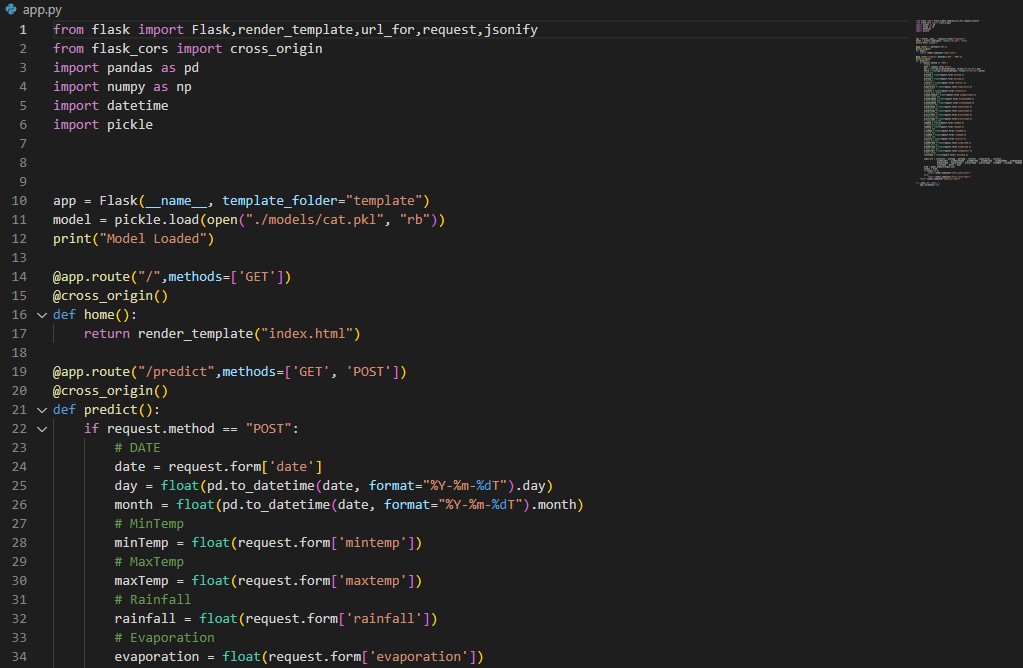
Source Code





Frontend

Figure 11: Landing Page





Backend

Figure 12: Flask

Seven Models

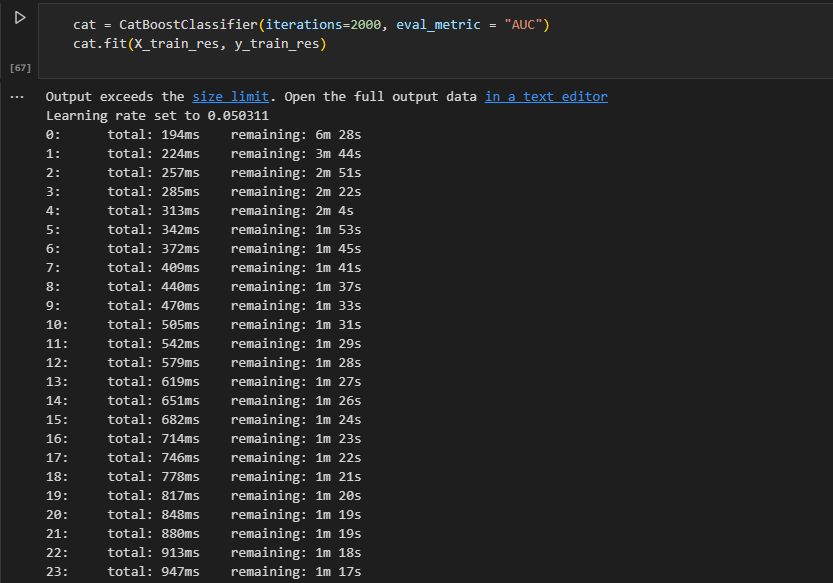


Figure 13 :catboost

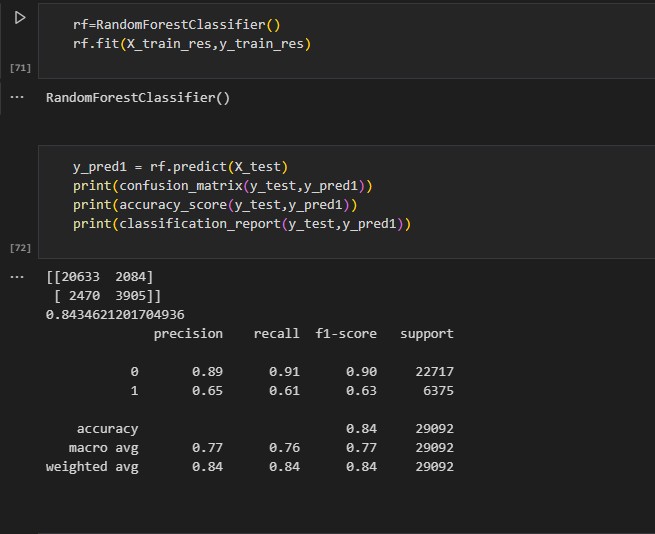


Figure 14 :random forest

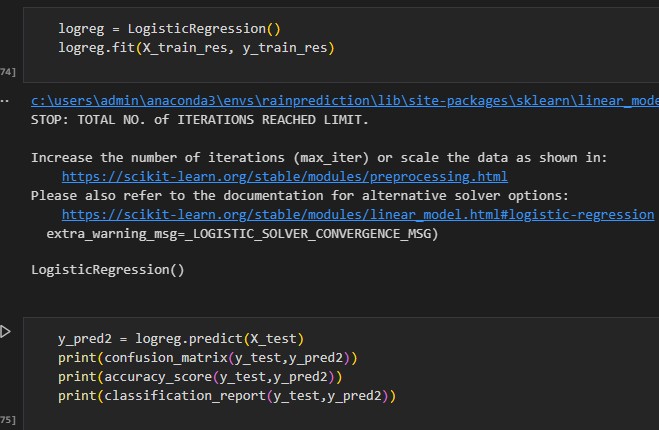


Figure 15 :logistic regression



Figure 16 :extreme gradient boosting

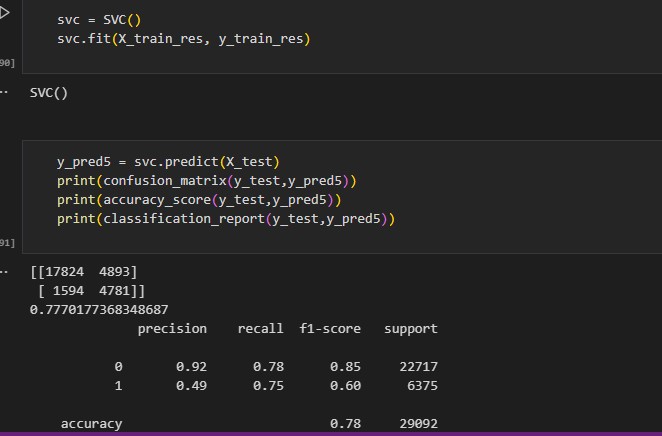


Figure 17:svm

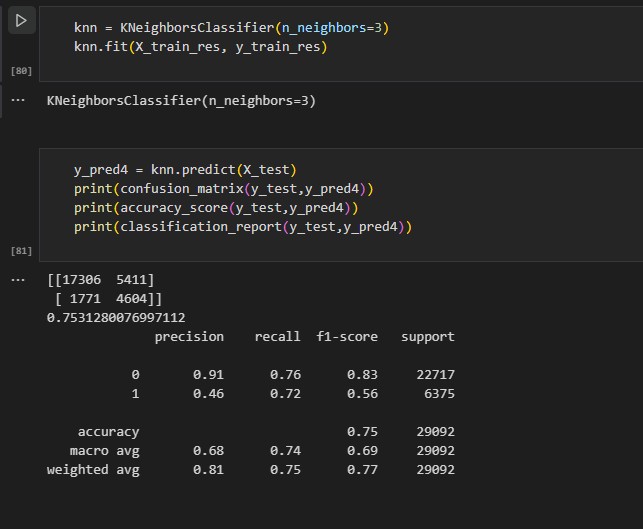


Figure 18 :k-nearest neighbors

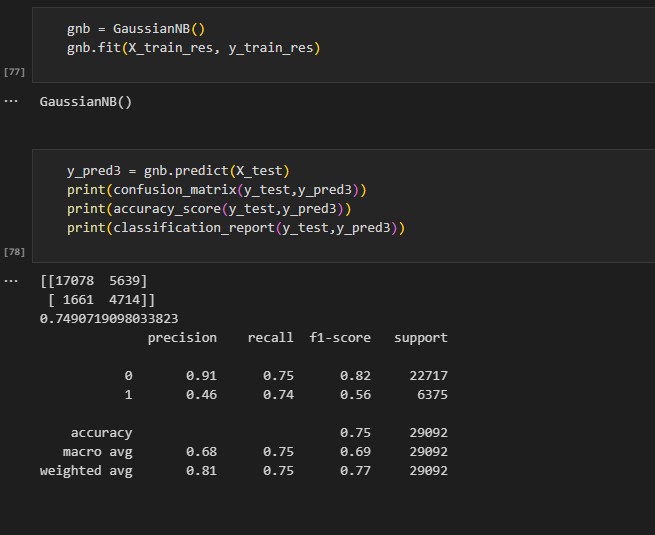


Figure 19: naïve bayes



Bibliography

1. “Shivam Tandon, Pawan Kumar Singh, Abhishek Patel” - Weather Prediction Using Machine Learning,2017.
2. “Suvendra Kumar Jayasingh, Jibendru Kumar Mantri, Sipali Pradhan” - Smart Weather Prediction Using Machine Learning, 2022.
3. “G. Hemalatha, K. Srinivasa Rao, D. Arun Kumar” - Weather Prediction using Advanced Machine Learning Techniques, AMSE 2021.
4. “Prathap N Kashyap, Preetham M S, Rohan Venkatesh Nayak, Uday B, Radhika T V” - Smart Weather Prediction Technique Using Machine Learning, 2020.
5. “Rajasekaran Meenal, Kiruthic Kailash, Prawin Angel Michael, Jeyaraj Jency Joseph, Francis Thomas Josh, Ekambaram Rajasekaran” - Machine learning based smart weather prediction, 2022.
6. “Amit Kumar Agarwal, Manish Shrimali, Sukanya Saxena, Ankur Sirohi, Anmol Jain” - Forecasting using Machine Learning, 2019.
7. “Rudransh Kush, Shubham Gautam, Sai Suvam Patnaik, Tanisha Chaudhary” - Climate Monitoring and Prediction using Supervised Machine Learning, 2022.
8. “Nazim Osman Bushara and Ajith Abraham” - Weather Forecasting in Sudan Using Machine Learning Schemes, 2014.
9. “Sana Khan, Mani Priya Mishra, Rukaiya Khatoon, Ritika Singh” - Weather Prediction Using Machine Learning, 2022.