A project report on

WEATHER PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

by

GORLA NANDINI

19BCE7704

VIT-AP UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

May 2023

A project report on

WEATHER PREDICTION USING MACHINE LEARNING

Submitted in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

by

GORLA NANDINI

19BCE7704

VIT-AP
UNIVERSITY

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

May 2023

DECLARATION

I hereby declare that the thesis entitled “WEATHER PREDICTION USING MACHINE LEARNING” submitted by GORLA NANDINI (19BCE7704), for the award of the degree of SCOPE OF COMPUTER SCIENCE AND ENGINEERING, VIT is a record of bonafide work carried out by me under the supervision of DR. Selva Kumar S

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Amaravati

Date:27/05/2023 Signature of the candidate

GORLA NANDINI

CERTIFICATE

This is to certify that the Senior Design Project titled "**WEATHER PREDICTION USING MACHINE LEARNING**" that is being submitted by **19CE7704** is in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. SELVA KUMAR S

Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner	**External Examiner**

Approved by

PROGRAM CHAIR	**DEAN**
B. Tech. CSE	School Of Computer Science

ABSTRACT

Weather prediction is used to predict the conditions of the environment for a given data. It is used in the application of data science and technology. Weather prediction is more helpful for people because it predicts how the future weather will be and people may plan accordingly. Farmers will be the most beneficial one’s as they may know the rainfall prediction and grow crops accordingly. The weather forecast can be done in many ways like using the history data or analyzing the current atmosphere. The authors predict the weather using the data of the attributes. The author used methodologies like Pre-processing, Data visualization, and some machine learning algorithms to predict the weather more accurately. Pre-processing is completed based on the data to fix the missing values and visualization is for the data representation in the form of graphical or image. In many fields of research and in industrial and military applications Digital-image processing has become economical.

Keywords:

Weather Prediction, Data Pre-Processing, Visualization of Data, Machine Learning Algorithm.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to my guide

Dr. Selva Kumar S, Assistant Professor (Sr), School of Computer Science & Engineering, VIT-AP, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Machine Learning.

I would like to express my gratitude to **G. Viswanathan, Sekar Viswanathan, Dr. S. V. Kota Reddy, and Dr. Madhusudhana Rao N, SCOPE,** for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Saroj Kumar Panigrahy. Associate Dean, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Amaravati	GORLA NANDINI
Date:27/5/2023	Name of the student

CONTENTS

CHAPTER 1

INTRODUCTION

CHAPTER 2

LITERATURE SURVEY

CHAPTER 3

METHODOLOGY

CHAPTER 4

EXPERIMENTAL ANALYSIS & RESULT

CHAPTER 5

CONCLUSION

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

KNN	K-NEAREST NEIGHBOUR
SVM	SUPPORT VECTOR MACHINE
GBC	GRADIENT BOOSTING CLASSIFICATION
XGB	EXTREME GRADIENT BOOSTING CLASSIFICATION
RFC	RANDOM FOREST CLASSIFICATION
ANN	ARTIFICIAL NUERAL NETWORK
GPR	GRADIENT PROCESS REGRESSION
ID_3	DECISION TREE REGRESSION

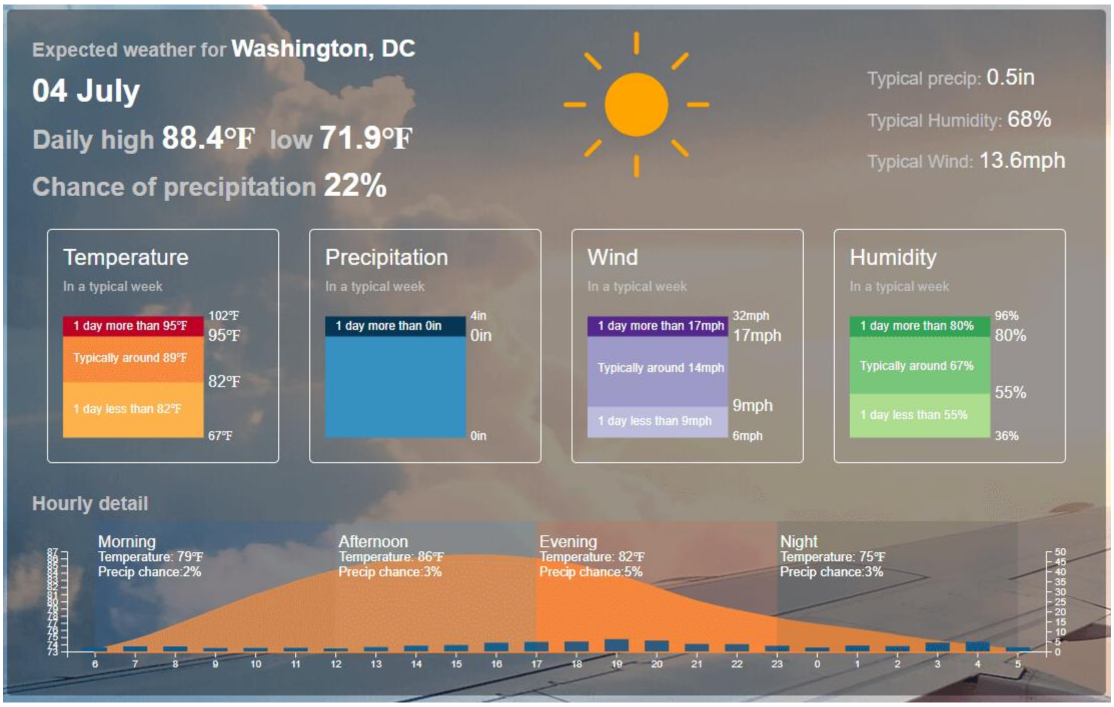
CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Weather prediction means awaiting the rainfall and telling how the rainfall changes with change in time and date. Change in rainfall occurs due to movement of land or transfer of energy. numerous meteorological patterns and features like rush, maximum temperature, minimal temperature, wind, and rainfall do due to the physical transfer of heat and humidity by convective processes. shadows are formed by evaporation of water vapor. As the water cycle keeps on evolving the water content in the shadows increases which in turn leads to rush. This is how the convective process happens and also the change in rainfall. numerous factors like temperature, downfall, pressure, moisture, sun, wind, and cloudiness are considered for prognosticating the rainfall. It's also possible to identify the different types of shadows associated with different patterns of rainfall. These patterns of rainfall help in prognosticating the rainfall cast.

In the history, people used barometric pressure, current rainfall conditions, and sky conditions to prognosticate whereas now there are numerous computer- grounded models that consider atmospheric factors to prognosticate the rainfall. These styles are not accurate and the reason is due to the chaotic nature of the atmosphere as it keeps on changing. Indeed, prognosticating the rainfall for a longer period of time won't be accurate that's why utmost of the current soothsaying (1) models prognosticate rainfall only for a couple of days not further than 10. The delicacy gets reduced with an increase in time.



Expected weather for **Washington, DC**

04 July

Daily high **88.4°F** low **71.9°F**

Chance of precipitation **22%**

Typical precip: 0.5in

Typical Humidity: 68%

Typical Wind: 13.6mph

Fig. 1 Historical Weather data to forecast

Weather prediction is not a pure process, and standard practices and procedures will be directly applied. Predictor’s job is predicated on theoretical background and lab work which needs study for more years but mainly day-to-day practice inside a weather prediction service having a particular technical environment. The work of the predicters has evolved significantly over the years to require the advantages of both scientific and technological improvements. The skill of numerical models has improved much a lot that some canter’s are automating routine forecasts to permit predictors to specialize in high-impact weather or areas where they can add significant value. So, it's dangerous to see a regular thanks to achieve weather predicts.

1.2 OVERVIEW OF MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI) which focuses on the development of algorithms and models that allow computers to learn and make predictions or decisions without being clearly programmed. It is based on the idea that machines can automatically learn from and analyse large amounts of data to identify patterns, extract insights, and make informed predictions or decisions.

The core concept behind machine learning is to enable computers to learn from data and improve their performance over time, without being explicitly programmed for every task or scenario. The learning process involves training a model on a labelled dataset, where the model learns to recognize patterns and relationships between input data and corresponding output or target values.

1.3 CHALLENGES PRESENT IN MACHINE LEARNING ALGORITHMS

While, machine learning algorithms have some powerful tools for solving complex problems, however they also face some challenges which are included:

1. QUALITY OF DATA: Machine learning algorithms heavily depend on the data quality. If the data has missing values or noisy or robustness, it can lead to poor performance. Data pre-processing techniques are necessary to solve these kinds of issues.
2. INSUFFICIENT DATA: Insufficient data can lead to overfitting, where the model performs well on the training data but fails to generalize too new.
3. OVERFITTING: It occurs when a model learns the training data too well, capturing noise and irrelevant patterns. This leads to poor generalization to new data.
4. MODEL INTERPRETABILITY: Many machine learning algorithms, such as deep neural networks, are often considered black boxes, making it challenging to interpret how they arrive at their predictions. This lack of interpretability can be problematic in sensitive domains where explanations are required.

5. COMPUTATIONAL COMPLEXICITY: Some machine learning algorithms which involve large dataset, can be computationally expensive and more time consuming. For this, training the model may require scalability to perform well.
6. CONCEPT DRIFT: In real-world applications, the data distributions may change over time. Models trained on historical data may become outdated and fail to adapt to new patterns. Continuously monitoring and updating models to handle concept drift is a challenge in dynamic environments.
7. SECURITY AND PRIVACY: Malicious actors manipulate input data to trick the model or cause incorrect predictions. Additionally, use of sensitive or personal data in machine learning systems raises concerns about privacy protection and data security.

Noticing these challenges requires research and development such as data collection and data pre-processing techniques areas in algorithm design.

1.4 PROJECT STATEMENT

As information gathered, we found that the different machine learning algorithms gave the best accuracy results for different dataset with different attributes. The current working raw dataset has been taken grounded on the whole datasets of different research papers which gave the best accurate values. So, on the grounded raw dataset some machine learning algorithms has applied on it which gave stylish accurate values which is called as best algorithms (stylish algorithms).

To predict the weather with given random data at any measure with the help of weather prediction which should certain any other conditions moreover drizzle, Fogg, rain, snow and sun. Even in rural areas where people don’t have access to internet should also be able to get weather forecast. To be able to solve the above problem, we are doing this project.

1.5 OBJECTIVES

The following are the objectives of this project:

- Some machine learning algorithms are used to train to find the weather condition for the future time where the random data was taken, rather than data in dataset.
- Machine Learning can be used to overcome disadvantages caused by overload data limitations.
- In this project, we are using supervised machine learning algorithms to predict the changes of environment or atmosphere.
- The main goal of this project is to predict the future condition of weather which helps us to plan accordingly. For example: Formers, Navy Officers and Astronaut’s etc...

1.6 SCOPE OF THE PROJECT

Data collection has been completed by using grounded datasets of different research papers. The dataset should be pre-processed to clean the data, fill in the missing values, and prepare the data for analysis which is suitable for all models. Among precipitation, maximum temperature, minimum temperature, wind, and the weather, weather attribute has been used as inputs for the machine learning model. This feature includes drizzle, Fogg, rain, snow and sun, and others.

The dataset has been trained for 80 Percentage (%) and tested for 20 Percentage (%). We have selected the best accuracy rate algorithms named as K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Gradient Boosting Classifier (GBC), Extreme Gradient Boosting Classifier (XGB), Random Forest Classification (RFC), and Artificial Neural Network (ANN). All algorithms gave the best accurate values near to the highest best accuracy values which is given by ANN. By using the ANN model, we are predicting the weather by giving random data.

1.7 ORGANIZATION OF THE THESIS

In this document, chapter 2 consists literature survey. The literature survey talks about the research done to work on the project. All the details about the papers, websites on which the research work is done in order to work on the project which is provided by the literature survey. In chapter 3, we discuss about the various methodologies used in this project and how it helps to fill the missing data in dataset. In chapter 4, the details about the analysis of the experiment are discussed. The experimental analysis includes system configuration like software requirements, sample code, result screenshots for a tested input data. In the next chapter we give the conclusion about the project and also provided information if the project can be implemented further or not. In the final chapter we provide all the references which helps to do refer for this project.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY (BACKGROUND STUDY)

In a literature survey, students or scholars analyse the data critically and compactly before exploration and literature related to a particular exploration problem and use them for their own exploration purpose. It helps students in understanding of new exploration and its connection to earlier work. The review of exploration emphasizes advancement in the field by conducting a critical analysis of being literature.

Weather forecasting means prediction of atmosphere condition. Weather forecasting is a task of predicting the state of environment at future time and a specific location. This has been done using physical simulations in which environment is modelled as a fluid. The current state of atmosphere is sampled and the future state of atmosphere is computed by numerically solving the equitation the equitation of fluid dynamics.

2.1.1 WEATHER FORECASTING USING MACHINE LEARNING

Still, the system of ordinary discrimination equations that govern this physical model is unstable under disquiet, and misgivings in the original measures of the atmospheric conditions and a deficient understanding of complex atmospheric processes circumscribe the extent of accurate rainfall soothsaying to a 10-day period, beyond which rainfall vaticinations are significantly unreliable.

Machine Learning, on the negative, is fairly robust to disquiet and doesn't bear a complete understanding of the physical processes that govern the atmosphere. thus, machine literacy may represent a feasible volition to physical models in rainfall soothsaying.

Two machine literacy algorithms were enforced direct retrogression and a variation of functional retrogression. A corpus of literal rainfall data for Stanford, CA was attained and used to train these algorithms. The input to these algorithms was the rainfall data of the once two days, which include the maximum temperature, minimal temperature (3), mean moisture, mean atmospheric pressure, and rainfall bracket for each day. The affair was also the outside and minimal temperatures for each of the coming seven days.

In this paper, details of rainfall for the history 2 days are considered. Those details are considered as input and performing direct retrogression and variation of functional retrogression, affair is attained. The affair is rainfall for coming 10 days. Generally, the bracket of rainfall gives 9 classes clear, haphazard shadows, incompletely cloudy, snow, thunder storm, rain, heavy, fog, substantially cloudy. The least mean square error for the direct retrogression and variation on functional retrogression is calculated and literacy angles are drawn in this paper.

Both direct retrogression and functional retrogression were outperformed by professional rainfall forecast services, although the difference in their performance dropped significantly for after days, indicating that over longer ages of time, our models may outperform professional bones. Linear retrogression proved to be a low bias, high friction model whereas functional retrogression proved to be a high bias, low friction model. Linear retrogression is innately a high friction model as it's unstable to outliers, so one way to perfect the direct retrogression model is by collection of further data.

Linear retrogression is low prejudiced with high friction model whereas functional is exactly contrary to it. Collection of further data can make better the direct retrogression model. Hence the author suggests to consider 4 to 5 days of data as input to the model. Functional retrogression, how- 5 ever, was high bias, indicating that the choice of model was poor, and that its prognostications cannot be bettered by farther collection of data. This bias could be due to the design choice to cast rainfall grounded upon the rainfall of the once two days, which may be too short to capture trends in rainfall that functional retrogression requires.

2.1.2 MACHINE LEARNING MODEL FOR WEATHER FORECASTING

The project is to predict the temperature using different supervised machine learning algorithms such as Linear Regression, Random Forest Classifier (RFC), and Decision Tree Regression (ID_3). The input values are taken as text and numerical. The output value should be numerically based on the multiple extra factors like consisting of attributes named as maximum Temperature, minimum temperature, cloud cover, humidity, and son hours in a day, precipitation, pressure and wind speed etc…

There are different methods for predicting the temperature by using Regression and a variety of Functional Regression, in which datasets are utilized for literature survey. If the dataset needs to be trained, the calculations 80 percentage (%) size of information is needed and it has been utilized and 20 percentage (%) of information is named as test data and it’s been used.

Still, India exercising these Machine Learning computations, we will use 8 Times of information to prepare the trains and 2 times of information as a Test, if we need to expect the temperature of Kanpur. The as defied to Weather predicting using Machine Learning Algorithms which depends basically on re-enactment dependent on drugs and Differential Equations which incorporates models, for illustration, Linear regression, ID_3, RFC.

Those algorithms were outperformed by professional weather forecast services. Linear Regression is a natural high difference model as which is unsteady of outliers, so one can do to improve the linear regression model which is by gathering more information whereas RFC is most popular regression model and it proves that it is most accurate regression model. Therefore, it has more tests for forecasting the accurate outcomes which are utilized in numerical ongoing frameworks like offices, air terminals and traveller’s and so on...

2.1.3 WEATHER FORECASTING USING MACHINE LEARNING TECHNIQUES

Weather forecasting is the operation of scientific approaches and technology to predict the conditions of the atmosphere at a certain position and time. Weather prediction in old time is carried out by hand, using changes in barometric pressure, current weather conditions, and sky condition or darkness cover, weather forecasting now relies on computer- based models that grip numerous atmospheric factors into account now relies on computer- grounded models that take numerous atmospheric factors into account.

For a long moment, the researcher had tested to establish a straight relationship between the input downfall data attributes and the corresponding target attribute. But the discovery of nonlinearity within different attributes of rainfall data, the focus has shifted towards the nonlinear forecast of the rainfall. Weather forecasts are framed by collecting quantitative data about the current state and preceding going of the atmosphere and using scientific understanding of atmospheric processes to predict how the atmosphere will develop. The rainfall warning is important for the protection of life and property. Rain prognostications can be used by growers. In order to break down how the different machine learning ways will perform in the prognosis of downfall. We've trained different types of machine literacy models on data collected from the field rainfall station of several metropolises.

Weather data is considered with different attributes for Weather forecast. The weather prediction experimentation was carried out to deconstruct the performance of different machine learning methods. We trained three different machine learning algorithms mentioned as SVM, ANN and time series RNN on this data. We also used these models to forecast weather and calculated root mean square error from the existent temperature. From observation of this blueprint, we set up out that time series using RNN is a better methodology for weather forecasting.

2.2 EXISTING SYSTEM

The existing system for weather prediction mainly uses dataset without missing values, noisy and robustness to predict the weather. Most of the websites which gave the best weather forecast with help of dataset as well as some machine learning technologies to forecast the weather. Very rarely found that applications or websites diagnosis the weather using satellite which we get the weather condition or any particular condition by using data they predict weather with the best accurate condition in the atmosphere. They track the movement of water vapour, precipitation and clouds and used that to forecast for the future time weather condition. This is the existing system for weather forecasting.

CHAPTER 3

METHODOLOGY

3.1 PROPOSED SYSTEM

Weather prediction is also done by using machine learning technologies but acquiring the weather dataset is a bit difficult and it would be tough having some particular attributes. Even forecasting using machine learning algorithms need more technologies and knowledge about algorithms. So, we are using machine learning technologies which process the best accurate values by pre-processing the data and by data visualization and machine learning algorithms.

3.1.1 ARCHITECTURE:

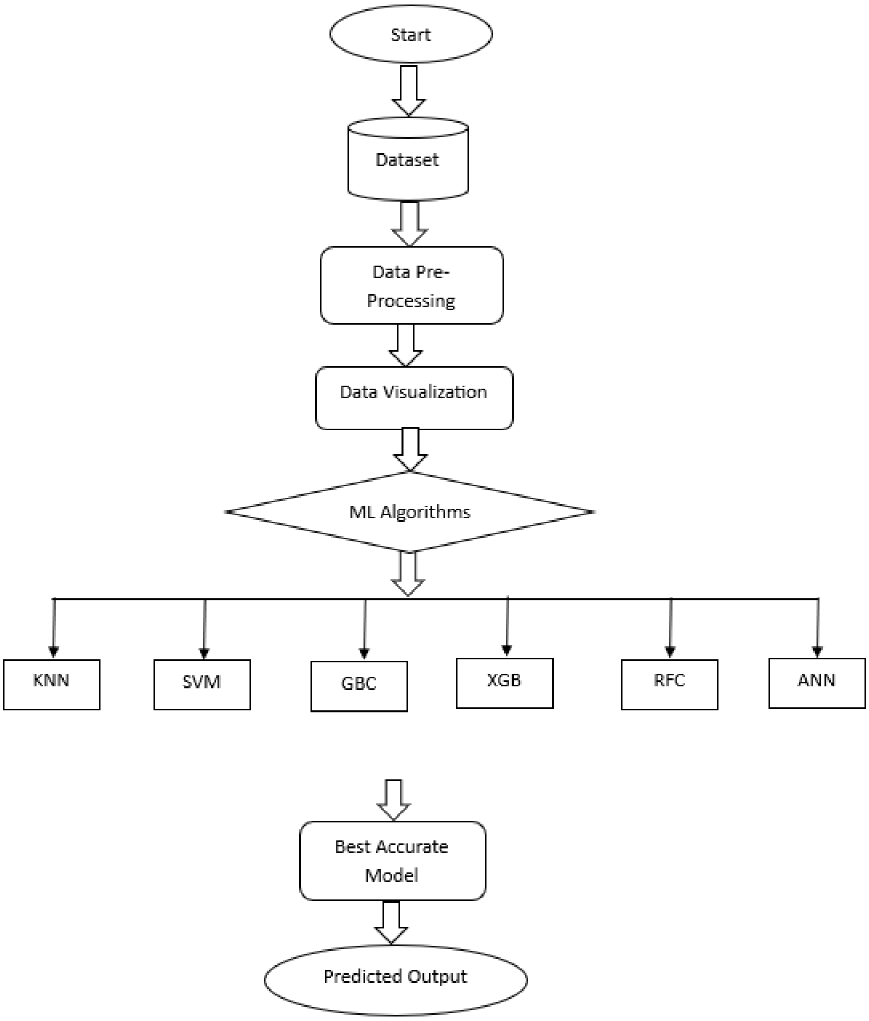


Fig. 2 Architecture

Every system or subject should be divided into modules for better understand and implementation. Dividing into modules helps the programmer and customer to work and use the system efficiently, independently. If any system isn't divided into modules and worked as a focused, also there comes a multiple error. Even we find difficulty in correcting those errors. It is must and should to disconnect the total project into modules and work on each and every module independently to get effective results. Our total system is divided into three modules namely:

- Pre-processing of dataset
- Data visualization
- Machine Learning Techniques
- Predicted Output

3.1.2 DETAILED DESIGN

Detailed design, by name itself saying that it's about the details of the project. It also provides a concise overview of projects purpose, recommendation and key findings. It also provides step by step process where architecture also gives, which helps the beginner at first sight just by looking.

A detailed design for a report should present information in a logical and coherent manner, making it easy for readers to understand the problem, grasp the findings, and take action based on the recommendations provided. Additionally, consider the target audience of the report and tailor the level of technical detail and language accordingly. Visuals should be well-designed and easy to interpret, supporting the narrative of the report.

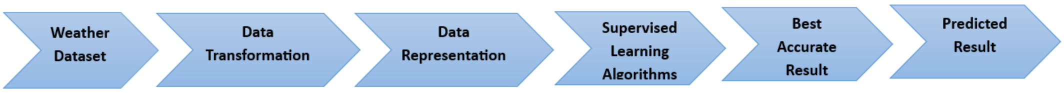


Fig. 3 Detailed Design

3.1.3 INTERFACE DESIGN

Interface design is referred as the process of designing the visual appearance, layout, and interaction of a software application, website, or any other digital product. It focuses on creating an intuitive and user-friendly interface that enables users to interact with the diagram and accomplish their tasks.

Interface design is an iterative process. It often incorporates user research, usability testing, and feedback gathering to continuously improve the design and align it with user needs and expectations.

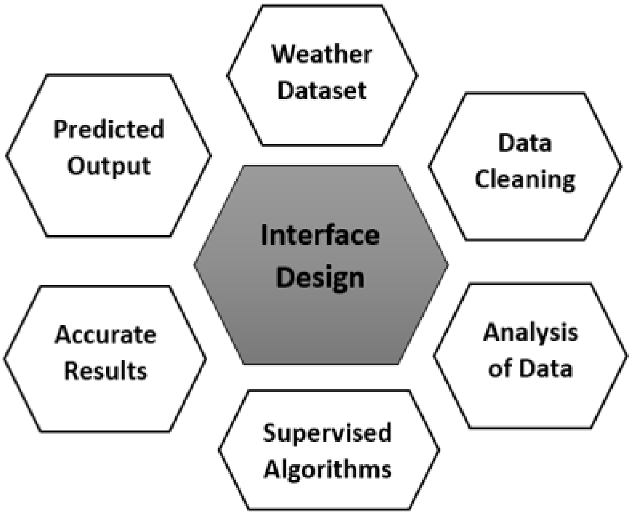


Fig. 4 Interface Design

3.2 PRE-PROCESSING OF DATASET

Data pre-processing is an essential step in data analysis and machine learning tasks. It involves transforming raw data into a format suitable for further analysis and modelling. The goal of data pre-processing is to enhance data quality, address missing or inconsistent values, reduce noise, and make the data more understandable and useful for the intended task.

3.2.1 DATA CLEANING:

This involves handling missing values, duplicate records, and correcting errors in the data. Missing values can be imputed using various techniques such as mean, median, mode, or predictive modelling. Duplicate records are typically removed to avoid bias in the analysis.

3.2.2 DATA NORMALIZATION:

Normalizing the data ensures that all variables are on a similar scale. It prevents certain features from dominating the analysis due to their larger magnitudes.

3.2.3 FEATURE SELECTION:

Feature selection involves identifying the most relevant features that contribute significantly to the analysis or modelling task. It helps reduce dimensionality and improves model performance. Techniques for feature selection include correlation analysis, information gain, and regularization methods.

3.2.4 FEATURE ENCODING:

Categorical variables need to be encoded into numerical representations for many machine learning algorithms. It includes common encoding techniques such as one-hot encoding, label encoding, and ordinal encoding.

3.2.5 DATA INTEGRATION: When working with multiple data sources, data integration is necessary to combine and merge data from different formats, structures, or databases.

The specific techniques and steps involved may vary depending on the dataset, the analysis goals, and the machine learning algorithms or analysis methods being used. It's important to consider the characteristics of the data and the requirements of the task to determine the most appropriate data pre-processing steps to apply.

3.3 DATA VISUALIZATION

Data visualization means the graphical representation of data and information using visual elements such as charts, graphs, maps, and other visuals. It is the process of presenting complex datasets in a visual and easily understandable format. The goal is to communicate information effectively, and relationships within the data.

Data visualization transfer the raw data into visual representations, it allows individuals to grasp the significance of the data more easily. It can reveal patterns, outliers, correlations, and other insights that might not be apparent when examining the raw data alone. Data visualization helps in identifying trends, making comparisons, understanding distributions, and presenting data-driven narratives.

Various types of visualizations can be used depending on the nature of the data. Data visualization include bar charts, line graphs, scatter plots, pie charts, histograms, heatmaps, tree maps, network diagrams, and geographical maps etc... The choice of visualization depends on the variables being analysed and the relationships between them. It simplifies the complex data and presents it in a visually appealing and understandable manner, aiding in data analysis, decision-making, and effective communication of insights.

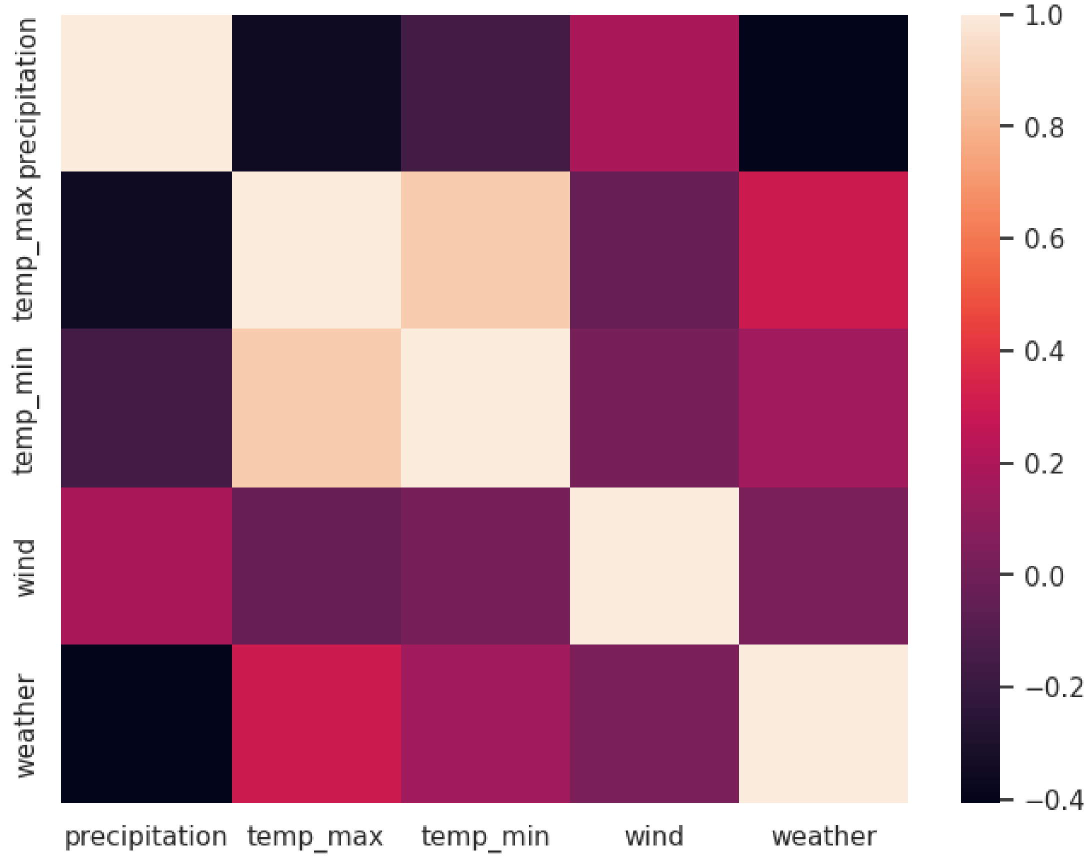


Fig. 5 Heatmap of Data Visualization

3.4 SUPERVISED MACHINE LEARNING ALGORITHMS

Supervised machine learning algorithms are a type of machine learning algorithms that learn from labelled training data to make predictions or decisions. In supervised learning, it provided with input data (features) and corresponding output labels (target variable) during the training phase. The goal is to learn a mapping function that can predict the correct output for new, unseen input data.

The process of supervised learning involves several key components such as Training Data which is labelled dataset which is used to train the algorithm to learn patterns and relationships between feature labels. Second one is Feature Extraction which pre-process the data and extract relevant features from input data and it involves the transforming raw data into a format which suits for any model to learn from. Third one is that Model Selection which encompasses various algorithms with its own strengths and limitation. Fourth one is Model Evaluation which is already trained, need to be evaluated to assess its performance and generalization ability. And last one is Model Deployment and Prediction which is already trained and evaluated, can be deployed to make predictions.

3.4.1 K- NEAREST NEIGHBOUR

K-NEAREST NEIGHBOURS is a simple and powerful too in supervised machine learning algorithms. It is used in both classification and regression tasks. And KNN doesn't make anu assumptions on it own about the underlying data distribution. The name its self's is saying is that finding the nearest data that considered when making a prediction.

KNN predicts the class label or value of a new data point based on the majority opinion or average of its nearest neighbours in the training data. The underlying assumption is that similar instances tend to have similar output labels. The effectiveness of KNN predictions depends on the choice of the distance metric, the value of K, and the characteristics of the training data. It is important to consider these factors and evaluate the performance of the algorithm using appropriate evaluation metrics to ensure accurate predictions.

The KNN algorithm can race with the most accurate models because it makes largely accurate prognostications. Therefore, we can use the KNN algorithm for operations that need high perfection but that do not need a natural- readable model. The quality of the prognostications depends on the distance measure. It relies on calculation of distances between dyads of records. The algorithm is used in category problems where training data are available with given target values (output labels).

KNN is one of the best algorithms in prediction model which helps by utilizing the labelled training data. It stores the training data which consists of input feature data which serves as a reference for making predictions. KNN calculate the distance between the new datapoint and all the instances in training data. It is used in prediction for classification which assigns the class label to the new data point based on majority vote of its KNN.

3.4.2 SUPPORT VECTOR MACHINE

Support Vector Machine is also a powerful supervised machine learning algorithm which is used for both classification and regression tasks. SVMs are effective in handling complex decisions boundaries and have been widely applied in various domains. However, SVM is primarily used for classification problems in machine learning.

The main objective of SVM is to find the Maximum Margin Classifier that separates the data points of different classes with the largest possible margin. By using the margin, SVM can achieve better generalization performance and robustness. And SVM contain Support Vectors which are the data points that lie closest to the decision boundary. And also, it has the aspect of Kernal Trick which can handle non linearly separable data by applying Kernal Stick. SVM contain maximum (achieve better generalization) margin which represents the distance between boundary and support vectors.

SVM has the ability to handle high-dimensional data, robustness to noise, and the potential to find global optima due to the convexity of the optimization problem. However, SVM is computationally expensive, particularly in dealing with large datasets. Additionally, choosing the appropriate kernel function and tuning the hyperparameters (e.g., C and kernel parameters) can require careful experimentation and cross-validation.

3.4.3 GRADIENT BOOSTING CLASSIFIER

GBC refers to Gradient Boosting Classifier, which is a machine learning algorithm used for classification tasks. It belongs to the family of ensemble methods and combines multiple weak classifiers (decision trees) to create a strong predictive model. GBC helps in predictions by leveraging the power of gradient boosting to improve accuracy and make robust predictions.

GBC stars the training by initial Weak Classifier on labelled training data which makes predictions but accuracy may be limited. Also, GBC works on the residual calculation which calculate the difference between the predicted labels of the initial weak classifier and the labelled training data. It builds a weak classifier to correct the errors of previous classifiers.

GBC combines the prediction of all weak classifiers using weighted voting. By giving more weight GBC assigns higher importance to accurate predictions and reduces influence of weaker classifiers. In GBC, the final prediction is obtained by aggregating the predictions of all weak classifiers by considering their weights.

GBC have some advantages which mainly helps in prediction model such as combining multiple weak classifiers on correcting the errors of the previous classifiers which leads to improving in accuracy and better predictive performance. And for robustness, it is less prone to overfitting compared to individual Decision Trees. It can handle both numerical and categorical features. It can handle imbalanced datasets and can able to adapt with different types of data distributions.

3.4.4 EXTREME GRADIENT BOOSTING CLASSIFIER

Extreme Gradient Boosting is famous machine learning algorithm that helps in making predictions. Its ensemble's the method that combines the predictions of multiple weak predictive models (such as decision trees) to create a stronger and more accurate model.

XGB utilize boosting technique, which sequentially trains weak models and combines them to create a strong predictive model (ID_3). Boosting reduces bias and variance, improving the overall prediction accuracy. It iteratively builds ID_3 learning from the mistakes of previous trees. It assigns higher weight to instances to enable the subsequent trees to focus on improving predictions.

XGB includes regularization techniques to prevent overfitting. Regularization helps control the complexity of the model and reduces the impact of noisy or irrelevant features, leading to better generalization and improved predictions on unseen data. It can automatically learn how to handle missing values during training process by reducing the need of pre-processing data.

XGB supports various optimization objectives, such as regression, classification, and etc… It provides flexibility in choosing the objective function based on the problem at hand, ensuring accurate predictions for different types of tasks. It uses Hyperparameter tunning to improve the model performance.

3.4.5 RANDOM FOREST CLASSIFICATION

Random Forest Classifier (RFC) is an ensemble learning algorithm that combines multiple decision trees to make predictions. Making interpreting of ID$_3$ is a bit easier to do but combining several ID$_3$ are tougher. RFC helps in the prediction process by leveraging the collective decisions of individual decision trees and providing robust and accurate predictions.

RFC helps to provide high prediction accuracy. It aggregates from multiple decision trees. It builds an ensemble of decision trees, by constructing multiple decision trees, RFC can capture different patterns and relationships present in the data. RFC is also having Voting Mechanism which helps to determine the final prediction. It is less prone to overfit as compared to decision trees. The robustness improves the reliability of predictions. It prevents overfitting to noise and focuses on the underlying patterns in the data.

RFC performs well with high dimensional datasets and can able to handle a large dataset as well as number of input features. It can provide a measure of feature importance which can help in feature selection and understanding the underlying data patterns. It is capable of handling missing values by considering alternative paths in trees. RFC is a versatile and powerful algorithm that helps in accurate prediction by leveraging the strengths of multiple decision trees.

3.4.6 ARTIFICIAL NUERAL NETWORK

Artificial neural network is also a type of machine learning algorithm influenced by the structures and functions of biological neural networks. ANNs are designed to learn and recognize patterns in data, making them well-suited for prediction tasks. It can provide an accurate prediction in wide range of applications.

ANN helps at recognizing the complex patterns and relationship in data. This capability is particularly useful for prediction tasks where patterns need to be identified and utilized. ANN is typically trained on labelled datasets where inputs are associated with familiar outputs. ANNs can automatically learn relevant features from raw data and it can discover and extract relevant information during the learning process.

ANNs aim to learn underlying patterns and trends in the data, allowing them to generalize and provide predictions beyond the training set. ANNs can adapt to changing data patterns and update their internal parameters accordingly. This adaptability allows ANNs to continue making accurate predictions even when the underlying patterns in the data evolve over time. By adjusting their weights and biases, ANNs can continuously update their prediction capabilities.

3.5 PREDICTED OUTPUT

After implementation of all Supervised Machine Learning Techniques with weather dataset, we get to know about their accuracy rates or percentages. By observing those accuracy percentages, we concluded that out of those one or two can gave the best accuracy results. So, we are predicting the output by using random raw dataset and by referring those particular algorithms which were gave best accuracy results. The predicting result will be the future time weather condition.

CHAPTER 4

EXPERIMENTAL ANALYSIS & RESULT

4.1 SYSTEM CONFIGURATION

4.1.1 SOFTWARE REQUIREMENTS TOOLS

1. Python :

   Python is a high- position programming language known for its plainness, readability, and versatility. It's extensively used in colourful disciplines, including web expansion, data analysis, artificial intellectuality, scientific computing, and more. Python's popularity continues to grow due to its versatility, ease of use, and strong community support. It provides a powerful and expressive programming environment for a wide range of applications, making it a popular choice among beginners and experienced developers alike.

   It has a clean and easy syntax which makes beginner-friendly and popular among developers. It is a line-by-line executer which is called as interpreted language. It has a huge number of libraries to improve performance and frameworks which provide pre inbuilt function and tools for various tasks. And it supports Object Oriented Language and it is an open-source language.

2. Googlecolab :

   Google Colab is a cloud-based integrated development environment (IDE) provided by Google. It allows users to write, run, and collaborate on Python code using a web browser without the need for any setup or installation on their local machines.

   Colab provides limited resources and has usage restrictions to ensure fair usage. Sessions may time out after a period of inactivity, and there are limitations on the maximum execution time and available memory. However, for most small to medium-sized projects, Colab provides an excellent platform for coding, experimenting, and collaborating on Python projects.

3. numpy :

   Numpy is an elementary library in python programming language for numerical computing. NumPy is a library for the Python programming language, adding support for large multi-dimensional arrays and matrices, along with a large collection of high- position fine functions to operate on these arrays. It provides an important array object and a collection of functions for efficiently manipulating and operating on arrays.

The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other formulators. In 2005, Travis Oliphant created NumPy by co-opting features of the contending Numarray into Numeric, with extensive variations. NumPy is open- source software and has multitudinous contributors.

4. matplotlib :

Matplotlib is a one of the popular Python libraries used for creating different visualizations. It provides a wide range of functions and classes for generating plots, charts, histograms, scatter plots, and more. It is used to for data visualization which allows us to create high-quality plots to explore and present the data visually. It can directly plot NumPy arrays, making it easy to visualize data stored in arrays or perform mathematical operations before plotting. The compatibility between NumPy and Matplotlib facilitates efficient data processing and plotting workflows.

Matplotlib supports interactivity in plots. You can add interactive features like zooming, panning, and tooltips to enhance the user experience. It forms the foundation of a larger ecosystem of visualization tools in Python. It serves as the backend for other libraries such as Seaborn, Plotly, and Pandas, which build on top of Matplotlib to provide higher-level functionality and specialized plot types.

5. pandas :

Pandas is a powerful open-source library in Python specifically designed for data manipulation, analysis, and exploration. It provides high-performance, easy-to-use data structures and data analysis tools, making it a valuable tool for working with structured and tabular data.

Pandas is used for tabular representation which introduces two primary data structures named as Series and Data Frame. A Series is a one-dimensional labelled array capable of holding any data type, while a Data Frame is a two-dimensional table with labelled axes. It has been used for dT cleaning and pre-processing which handles the missing and noisy data in dataset.

6. seaborn :

Seaborn is a popular Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating aesthetically pleasing and informative statistical graphics. Seaborn enhances the visual appeal of plots while simplifying the creation of complex visualizations.

Seaborn is a valuable tool for data visualization in Python, particularly for statistical analysis and exploratory data visualization. Its integration with Pandas, improved aesthetics, and specialized plotting functions makes it a popular choice for creating informative and visually appealing visualizations.

7. scipy :

   Scipy is a powerful open-source library in Python that is used for scientific computing and technical computing. It provides a wide range of functions and algorithms for numerical integration, optimization, interpolation, linear algebra, signal processing, statistics, and more.

   Scipy provides a comprehensive set of tools and algorithms for scientific and technical computing tasks. Its extensive functionality, integration with NumPy, and compatibility with other scientific libraries make it a valuable resource for researchers, scientists, engineers, and data analysts working in various domains.

8. sklearn :

   The sklearn (Scikit-learn) library is a widely used open-source machine learning library in Python. It provides a comprehensive set of tools for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

   Scikit-learn is a valuable tool for both beginners and experienced practitioners in machine learning. Its ease of use, comprehensive functionality, and integration with other Python libraries make it a popular choice for a wide range of machine learning tasks and applications.

   Scikit-learn have some classes which are available in python and those are used as libraries too in this project named as StandardScalar, LabelEncoder, train_test_split, KNeighborsClassifier, SVC, GradientBoostingClassifier, XGClassifier, accuracy_score, Classification_report, confusion_matrix etc…

9. missingo :

   The term "Missingo" typically refers to a glitch in the original Pokémon games. However, if you are referring to a Python library named "Missingo," I'm sorry to inform you that there is no widely known or official Python library by that name as of my knowledge cut-off in September 2021.

   It's possible that a library with that name has been developed independently by someone, but it may not be widely recognized or supported. If you have more information or specific requirements related to a "Missingo" library, please provide further details so that I can assist you better.

10. warnings :

    The warnings library in Python provides a way to handle warning messages generated during the execution of a program. It allows developers to control how warnings are displayed, logged, or ignored, depending on their needs.

    The warnings library helped to best practices and follows python guidelines. It can notify you about these changes, allowing you to update your code accordingly and maintain compatibility with different Python environments. And some frame works may define their own custom warnings to convey specific information to handle unique scenarios.

    Remember that while warnings can be useful during development and debugging, it's generally recommended to address and resolve the underlying issues that generate warnings rather than ignoring or suppressing them in production code.

11. tensorflow :

    The TensorFlow library is a popular and powerful open-source machine learning framework developed by Google. It provides a complete set of tools and functionalities for structure and fixing machine learning models.

    The tensorflow library offers a high-level API called Keras which simplifies the process of training neural network, and supports Artificial Neural Network (ANN). It allows you to export trained models in a format suitable for deployment on various platforms, including mobile devices, web applications, and cloud environments

12. keras :

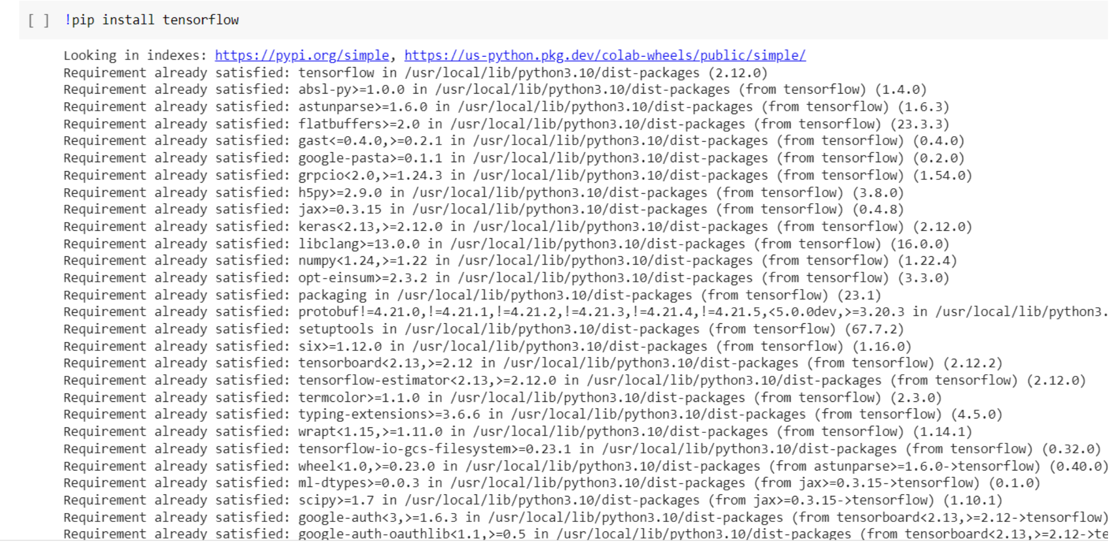
    The Keras library is a high-level neural networks API written in Python. It is designed to be user-friendly, modular, and extensible, making it a popular choice for building and training deep learning models. It offers tools for visualizing model architectures, enabling you to gain insights into the structure and parameters of your models.

    Keras provides a straightforward and easy-to-use API for defining and training neural networks. Keras supports a wide range of neural network architectures which provides pre-defined layers for building common types of networks, such as fully connected networks, Artificial Neural Network (ANN).

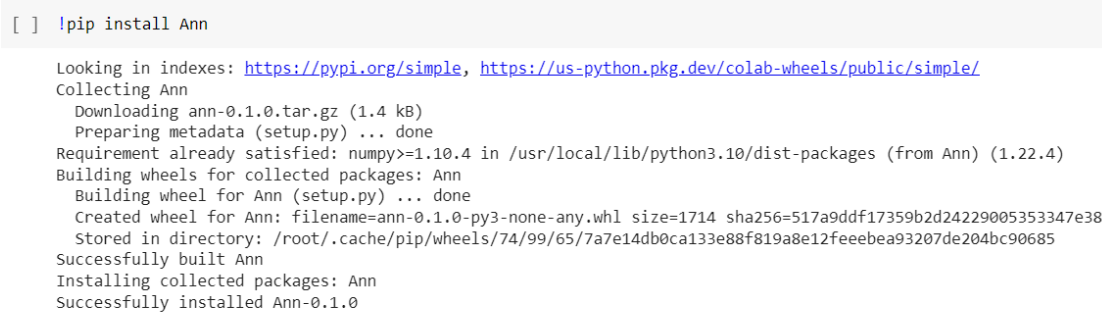
    With Keras, you can define your model architecture, compile it with an optimizer and loss function, and then train the model on your data using the fit() function. You can also evaluate the model's performance on test data and make predictions using the trained model. Keras provides extensive documentation and examples on the official TensorFlow website, which you can refer to for detailed usage instructions and tutorials.

4.2 SAMPLE CODE ALONG WITH OUTPUT SCREENSHOTS

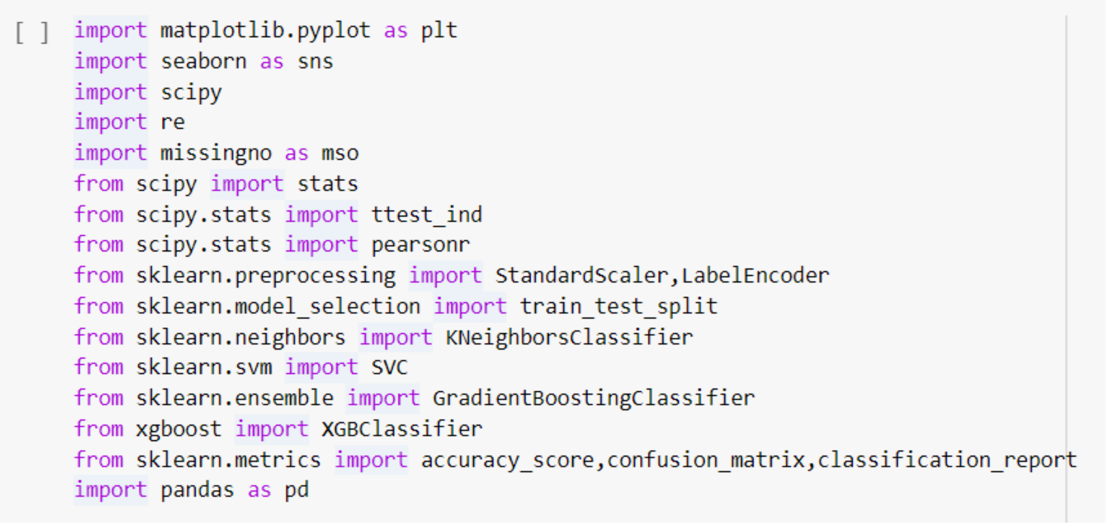
4.2.1 BASIC FUNCTIONS



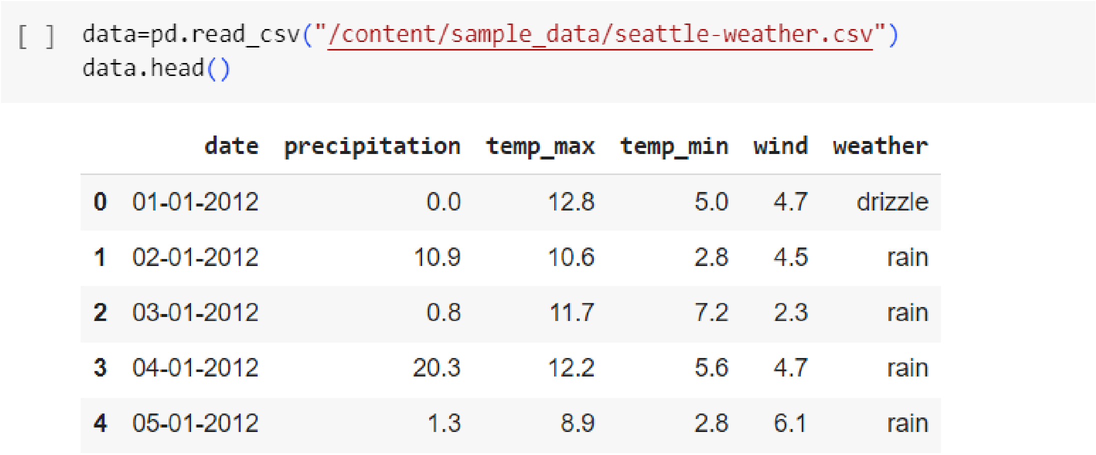
```
[ ]  !pip install tensorflow

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.12.0)
     Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
     Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
     Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.3.3)
     Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.0)
     Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
     Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.54.0)
     Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.8.0)
     Requirement already satisfied: jax>=0.3.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.8)
     Requirement already satisfied: keras<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.0)
     Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (16.0.0)
     Requirement already satisfied: numpy<1.24,>=1.22 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.22.4)
     Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
     Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.1)
     Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.
     Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
     Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
     Requirement already satisfied: tensorboard<2.13,>=2.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.2)
     Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.12.0)
     Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.3.0)
     Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.5.0)
     Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
     Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.32.0)
     Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.40.0)
     Requirement already satisfied: ml-dtypes>=0.0.3 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow) (0.1.0)
     Requirement already satisfied: scipy>=1.7 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow) (1.10.1)
     Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow)
     Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->te
```



```
[ ]  !pip install Ann

     Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
     Collecting Ann
       Downloading ann-0.1.0.tar.gz (1.4 kB)
       Preparing metadata (setup.py) ... done
     Requirement already satisfied: numpy>=1.10.4 in /usr/local/lib/python3.10/dist-packages (from Ann) (1.22.4)
     Building wheels for collected packages: Ann
       Building wheel for Ann (setup.py) ... done
       Created wheel for Ann: filename=ann-0.1.0-py3-none-any.whl size=1714 sha256=517a9ddf17359b2d24229005353347e38
       Stored in directory: /root/.cache/pip/wheels/74/99/65/7a7e14db0ca133e88f819a8e12feeebea93207de204bc90685
     Successfully built Ann
     Installing collected packages: Ann
     Successfully installed Ann-0.1.0
```

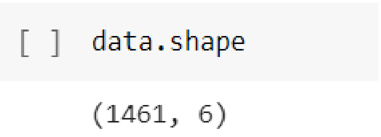


```
[ ]  import matplotlib.pyplot as plt
     import seaborn as sns
     import scipy
     import re
     import missingno as mso
     from scipy import stats
     from scipy.stats import ttest_ind
     from scipy.stats import pearsonr
     from sklearn.preprocessing import StandardScaler,LabelEncoder
     from sklearn.model_selection import train_test_split
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.svm import SVC
     from sklearn.ensemble import GradientBoostingClassifier
     from xgboost import XGBClassifier
     from sklearn.metrics import accuracy_score,confusion_matrix,classification_report
     import pandas as pd
```

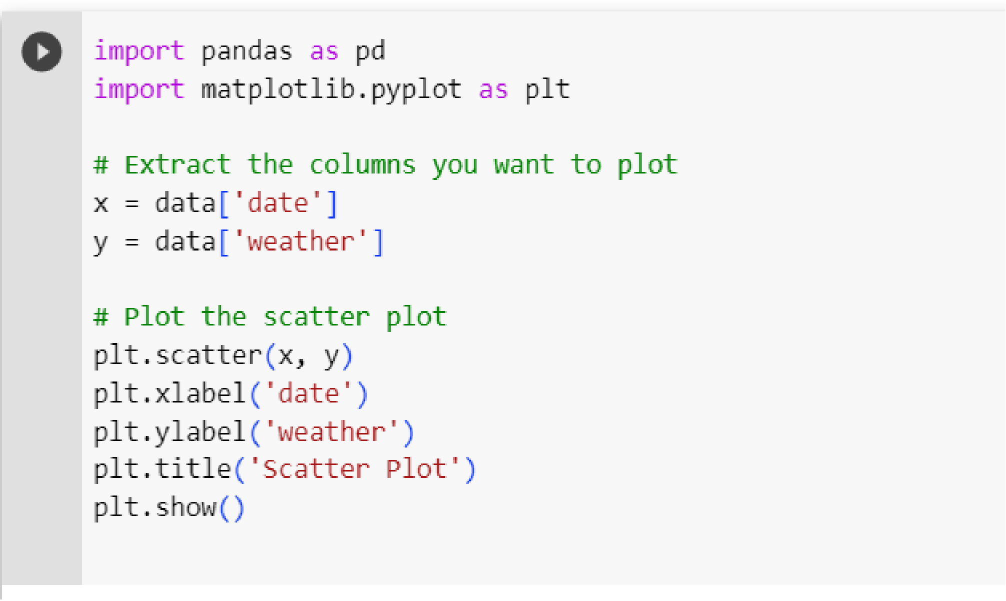


```
[ ] data=pd.read_csv("/content/sample_data/seattle-weather.csv")
    data.head()
```

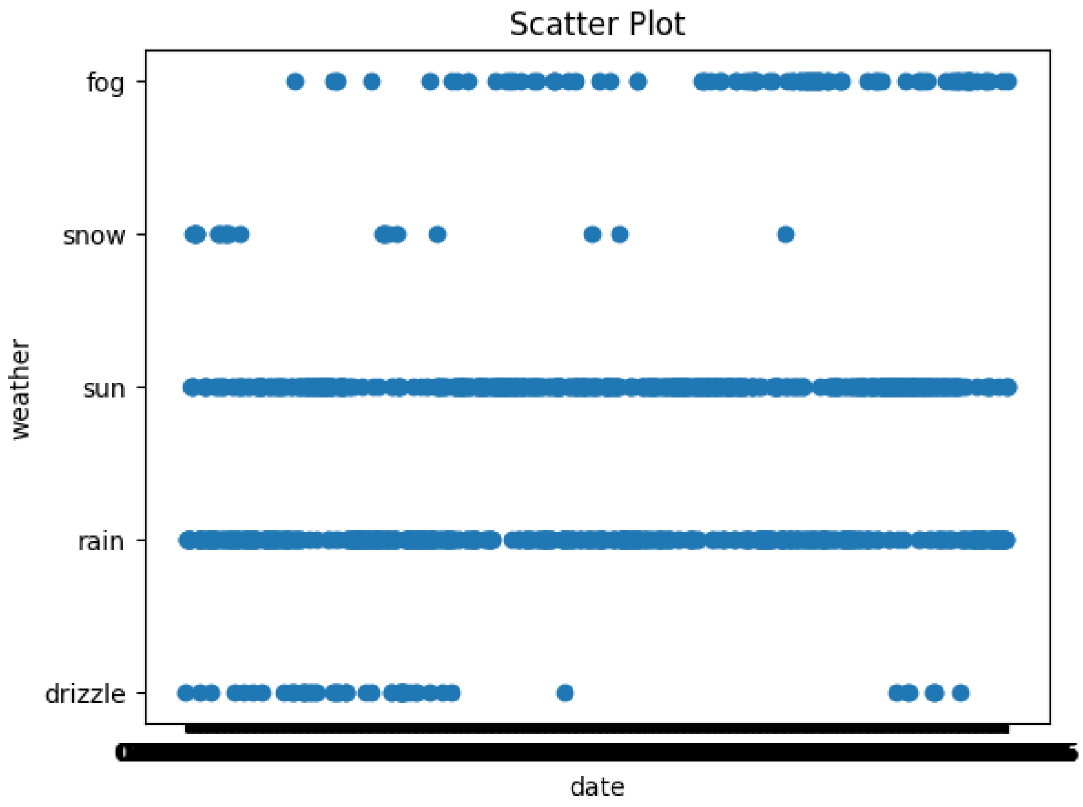
	date	precipitation	temp_max	temp_min	wind	weather
0	01-01-2012	0.0	12.8	5.0	4.7	drizzle
1	02-01-2012	10.9	10.6	2.8	4.5	rain
2	03-01-2012	0.8	11.7	7.2	2.3	rain
3	04-01-2012	20.3	12.2	5.6	4.7	rain
4	05-01-2012	1.3	8.9	2.8	6.1	rain

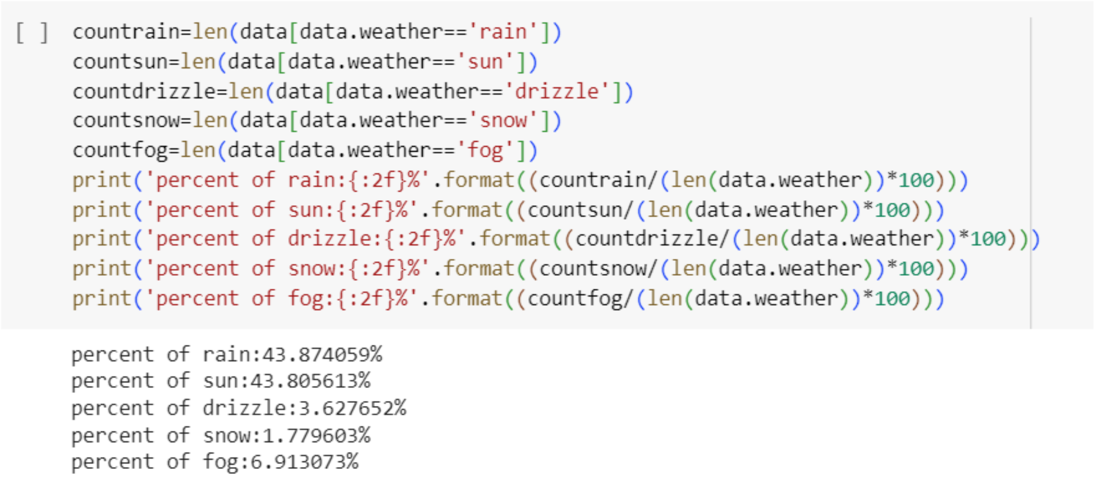


```
[ ] data.shape

    (1461, 6)
```



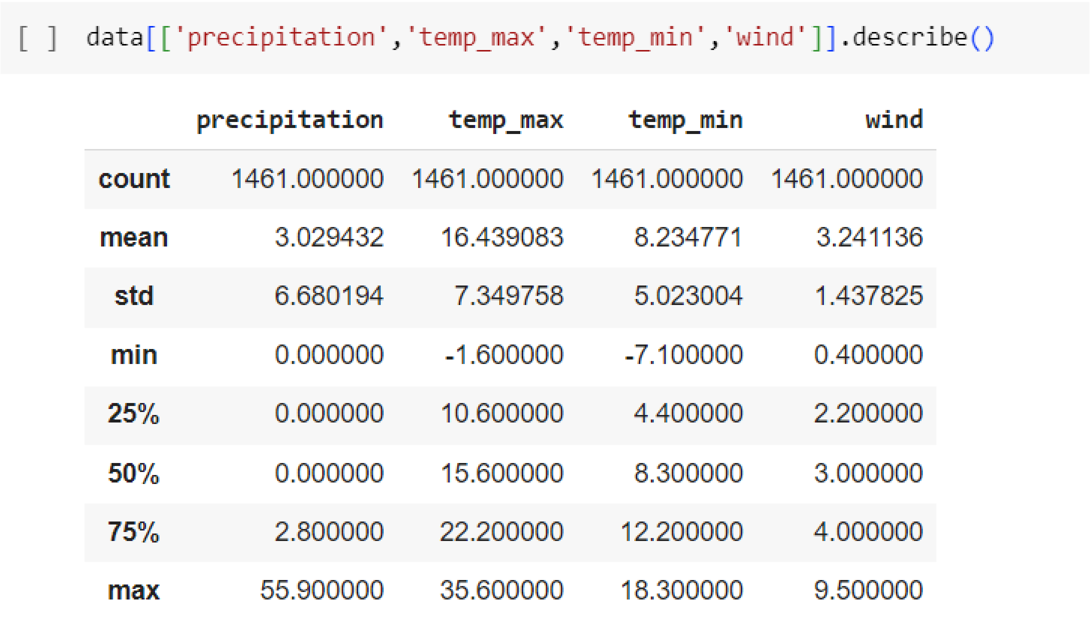
```
import pandas as pd
import matplotlib.pyplot as plt

# Extract the columns you want to plot
x = data['date']
y = data['weather']

# Plot the scatter plot
plt.scatter(x, y)
plt.xlabel('date')
plt.ylabel('weather')
plt.title('Scatter Plot')
plt.show()
```





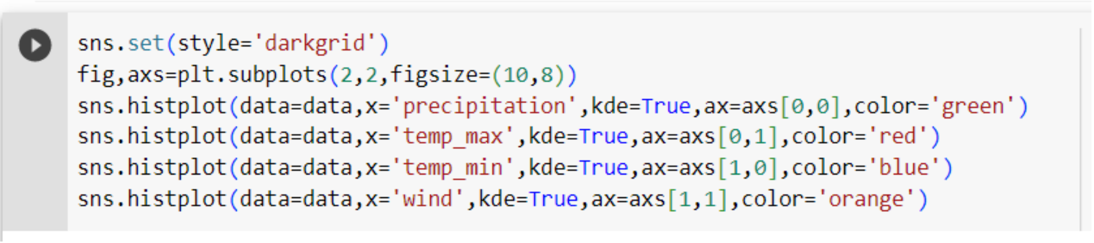
```
[ ]  countrain=len(data[data.weather=='rain'])
     countsun=len(data[data.weather=='sun'])
     countdrizzle=len(data[data.weather=='drizzle'])
     countsnow=len(data[data.weather=='snow'])
     countfog=len(data[data.weather=='fog'])
     print('percent of rain:{:2f}%'.format((countrain/(len(data.weather))*100)))
     print('percent of sun:{:2f}%'.format((countsun/(len(data.weather))*100)))
     print('percent of drizzle:{:2f}%'.format((countdrizzle/(len(data.weather))*100)))
     print('percent of snow:{:2f}%'.format((countsnow/(len(data.weather))*100)))
     print('percent of fog:{:2f}%'.format((countfog/(len(data.weather))*100)))
```

```
percent of rain:43.874059%
percent of sun:43.805613%
percent of drizzle:3.627652%
percent of snow:1.779603%
percent of fog:6.913073%
```

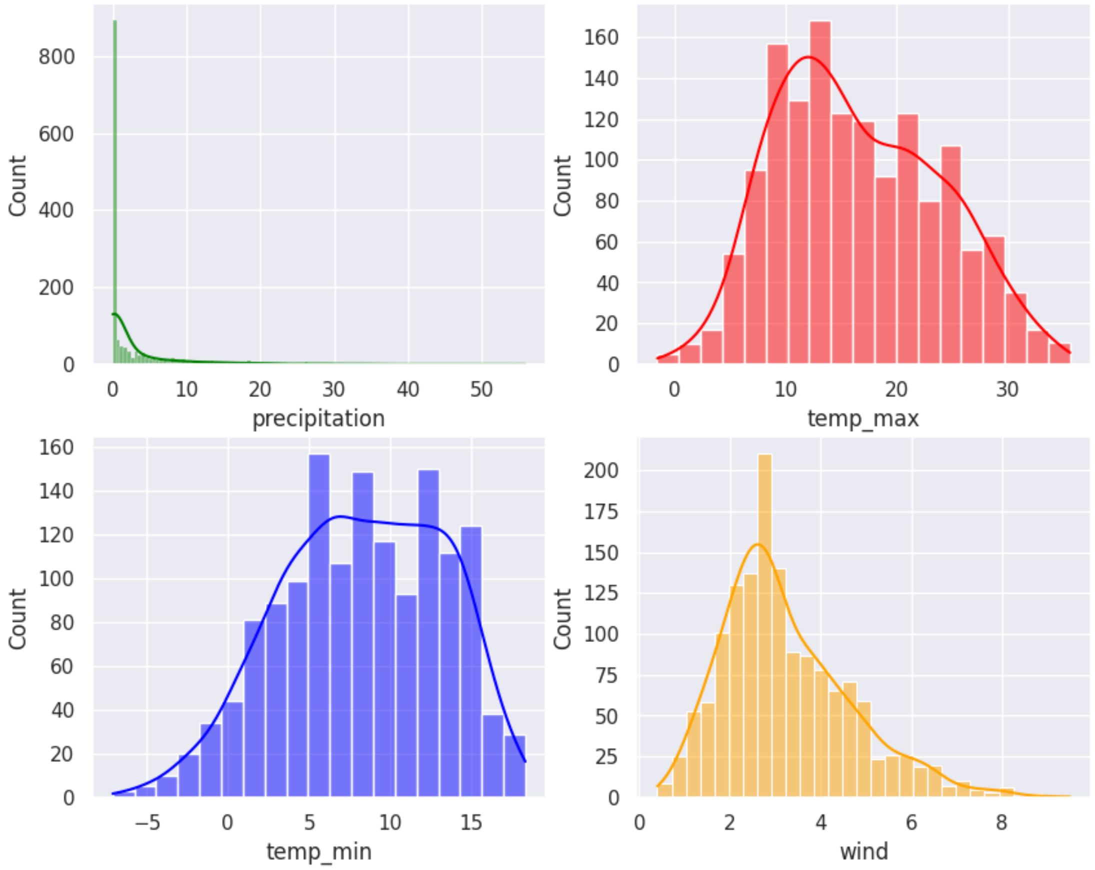


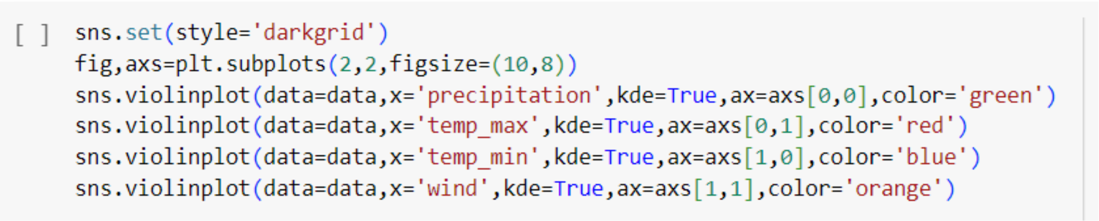
```
[ ]  data[['precipitation','temp_max','temp_min','wind']].describe()
```

	precipitation	temp_max	temp_min	wind
count	1461.000000	1461.000000	1461.000000	1461.000000
mean	3.029432	16.439083	8.234771	3.241136
std	6.680194	7.349758	5.023004	1.437825
min	0.000000	-1.600000	-7.100000	0.400000
25%	0.000000	10.600000	4.400000	2.200000
50%	0.000000	15.600000	8.300000	3.000000
75%	2.800000	22.200000	12.200000	4.000000
max	55.900000	35.600000	18.300000	9.500000

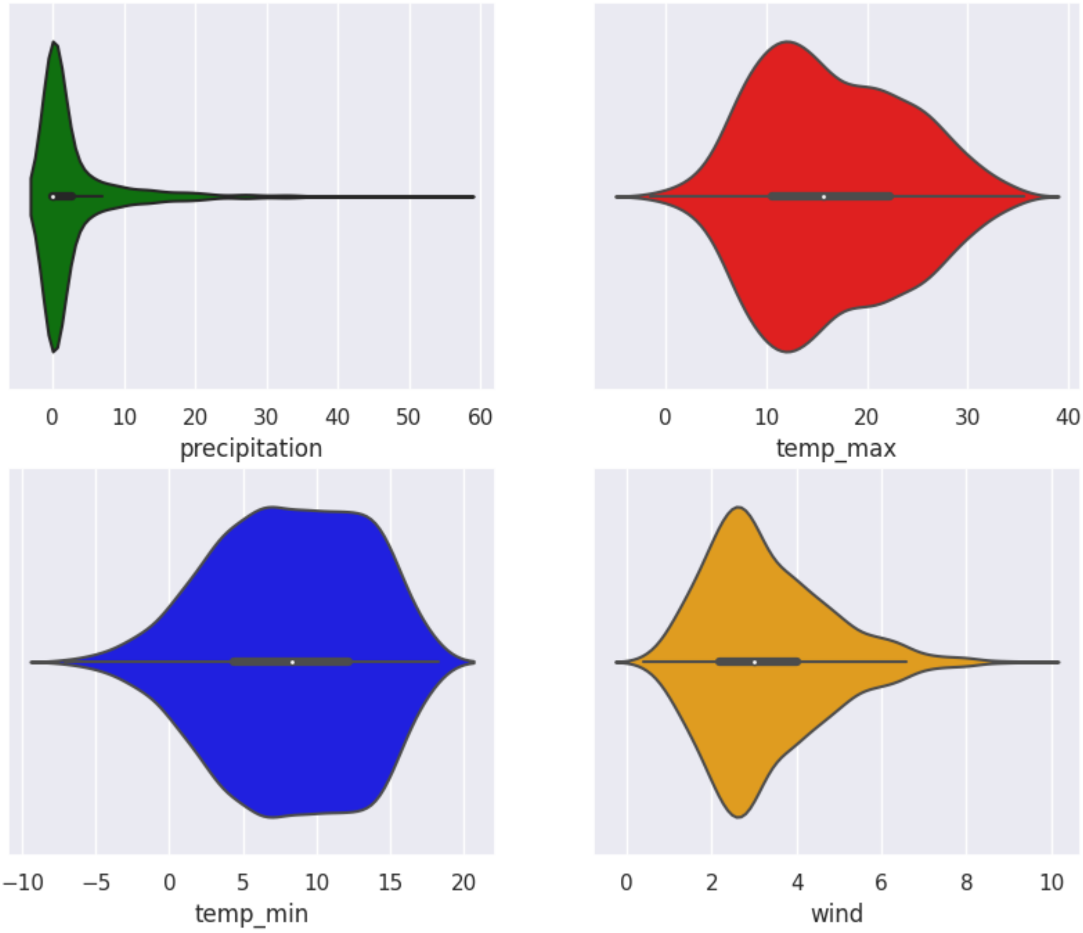


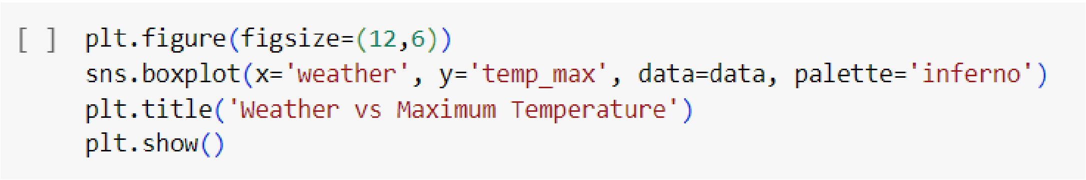
```
sns.set(style='darkgrid')
fig,axs=plt.subplots(2,2,figsize=(10,8))
sns.histplot(data=data,x='precipitation',kde=True,ax=axs[0,0],color='green')
sns.histplot(data=data,x='temp_max',kde=True,ax=axs[0,1],color='red')
sns.histplot(data=data,x='temp_min',kde=True,ax=axs[1,0],color='blue')
sns.histplot(data=data,x='wind',kde=True,ax=axs[1,1],color='orange')
```



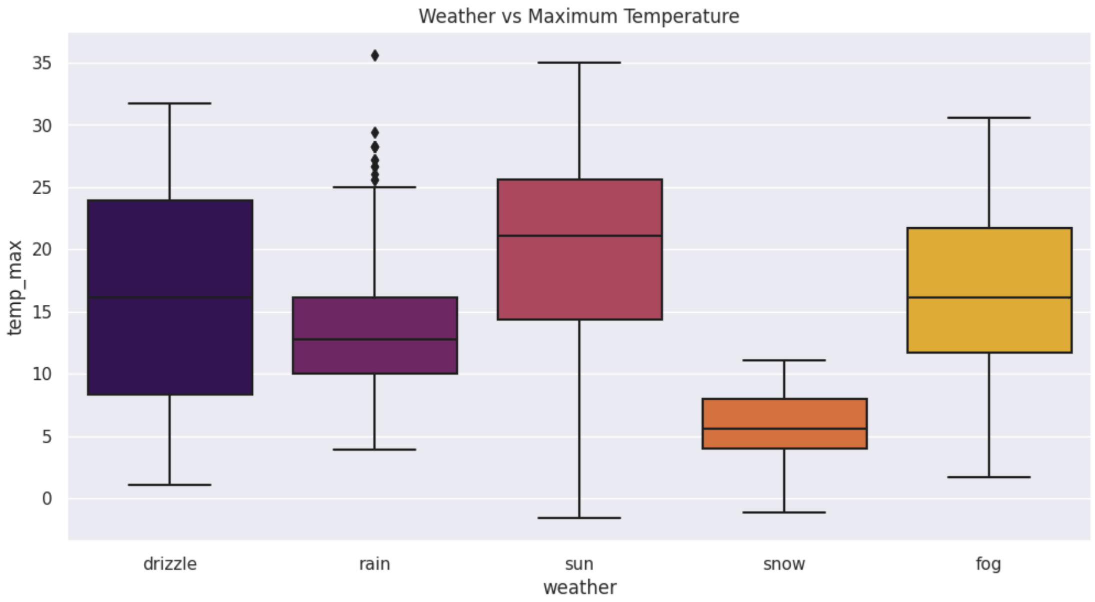


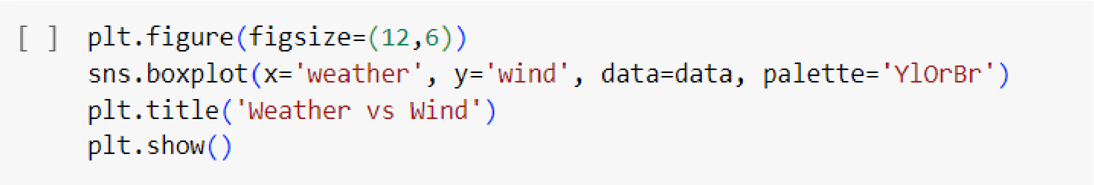
```
[ ]  sns.set(style='darkgrid')
     fig,axs=plt.subplots(2,2,figsize=(10,8))
     sns.violinplot(data=data,x='precipitation',kde=True,ax=axs[0,0],color='green')
     sns.violinplot(data=data,x='temp_max',kde=True,ax=axs[0,1],color='red')
     sns.violinplot(data=data,x='temp_min',kde=True,ax=axs[1,0],color='blue')
     sns.violinplot(data=data,x='wind',kde=True,ax=axs[1,1],color='orange')
```



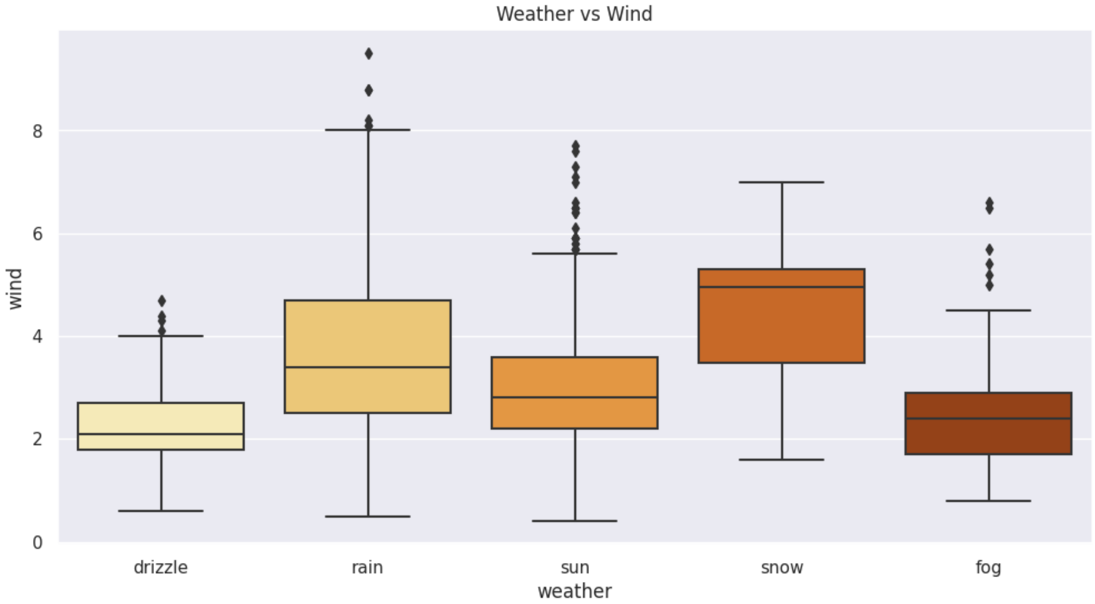


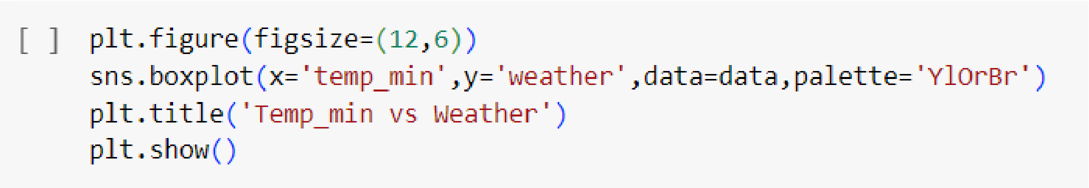
```
[ ]  plt.figure(figsize=(12,6))
     sns.boxplot(x='weather', y='temp_max', data=data, palette='inferno')
     plt.title('Weather vs Maximum Temperature')
     plt.show()
```



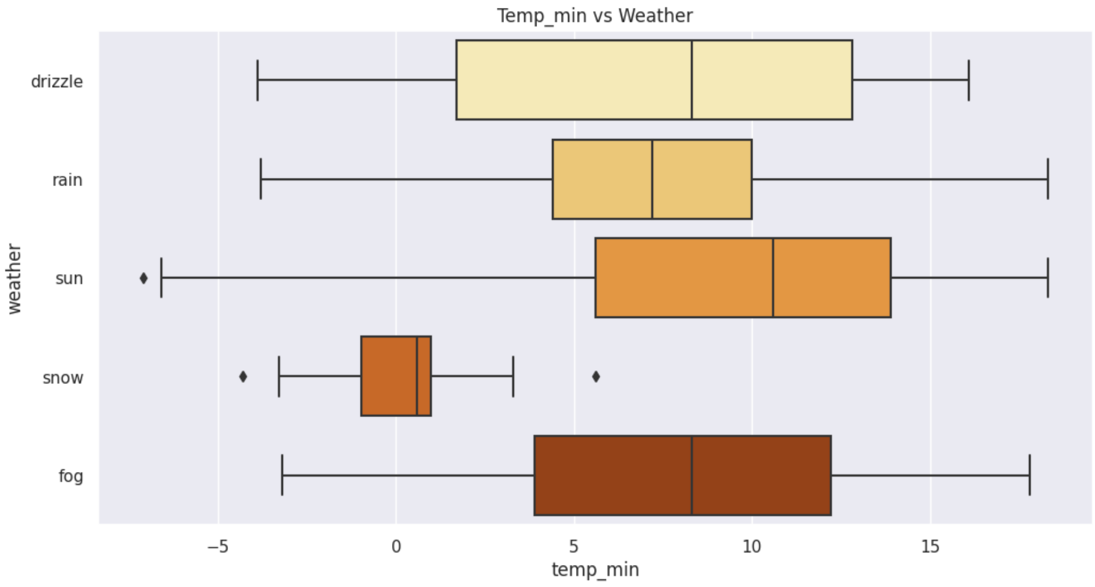


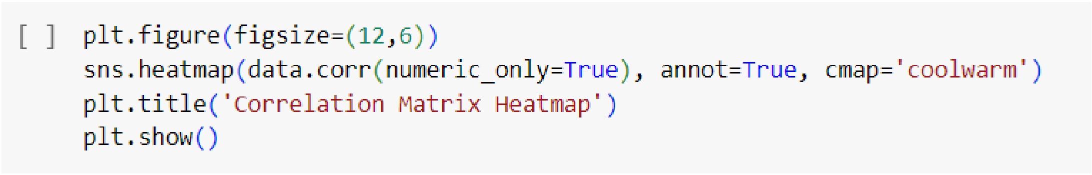
```
[ ]  plt.figure(figsize=(12,6))
     sns.boxplot(x='weather', y='wind', data=data, palette='YlOrBr')
     plt.title('Weather vs Wind')
     plt.show()
```



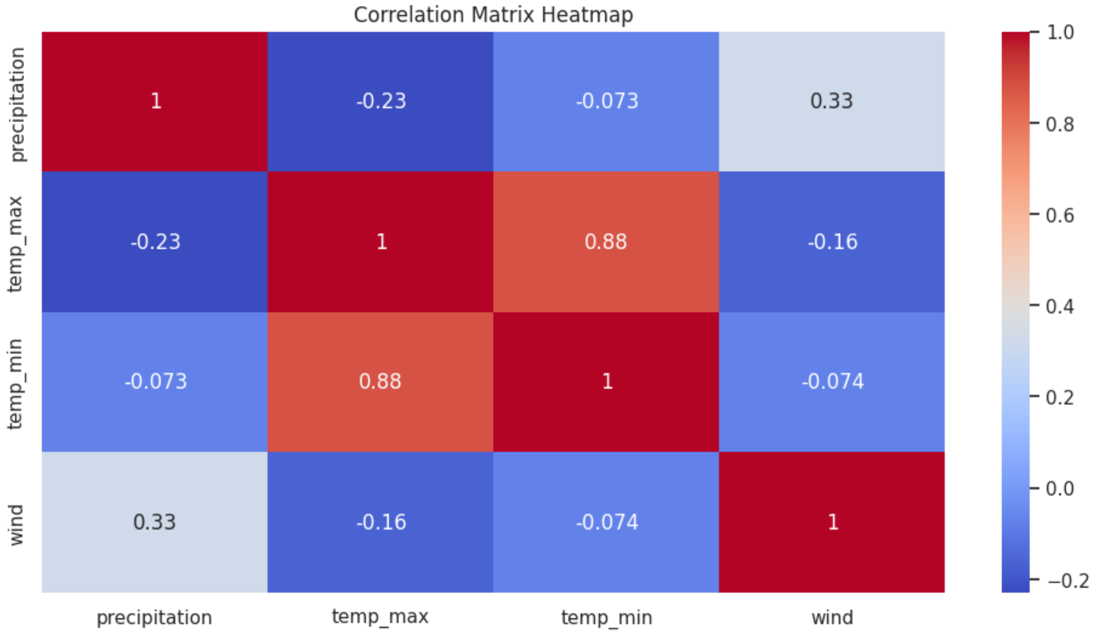


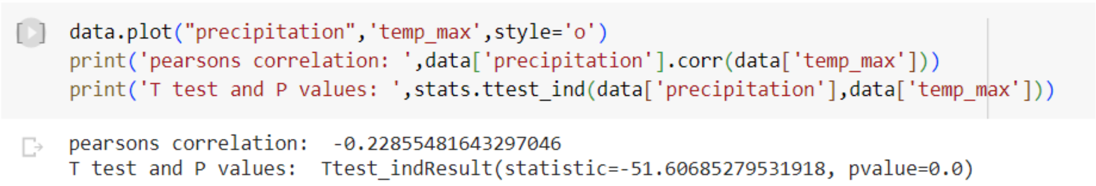
```
[ ]  plt.figure(figsize=(12,6))
     sns.boxplot(x='temp_min',y='weather',data=data,palette='YlOrBr')
     plt.title('Temp_min vs Weather')
     plt.show()
```





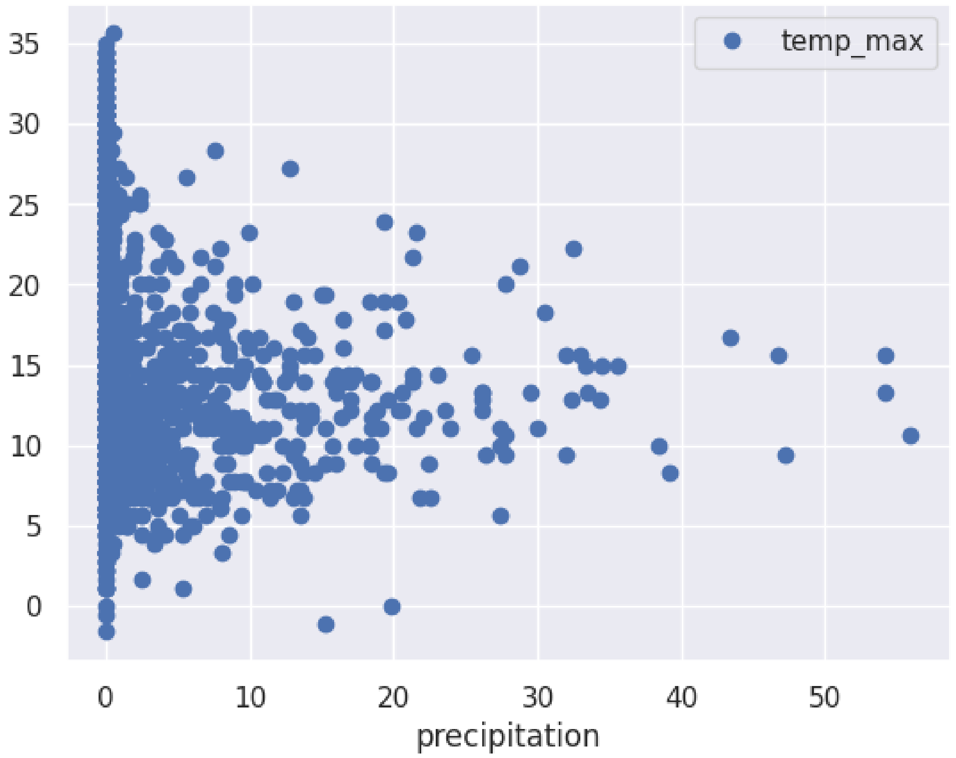
```
[ ]  plt.figure(figsize=(12,6))
     sns.heatmap(data.corr(numeric_only=True), annot=True, cmap='coolwarm')
     plt.title('Correlation Matrix Heatmap')
     plt.show()
```

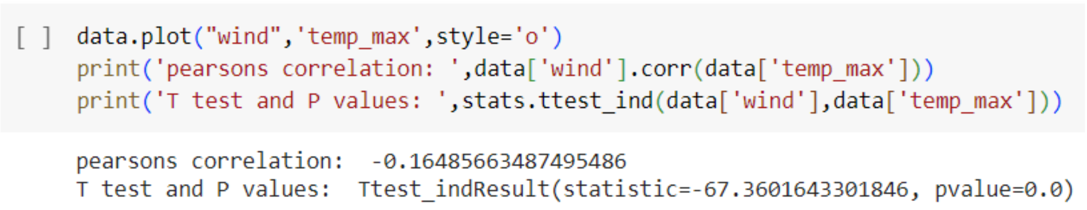




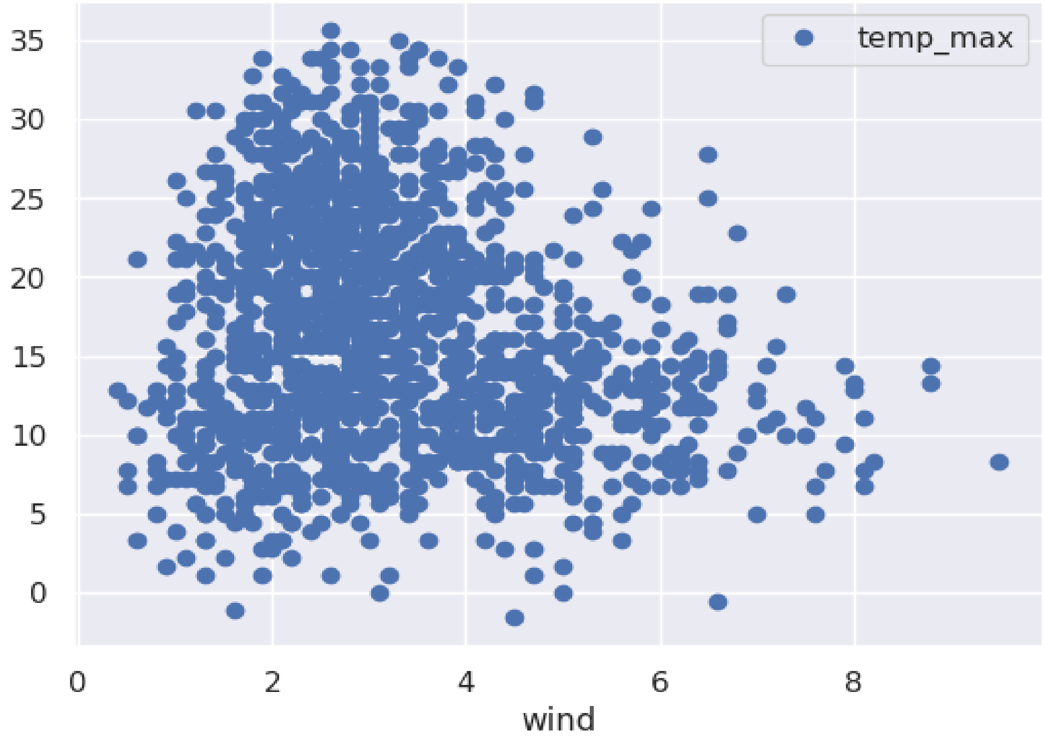
```
data.plot("precipitation",'temp_max',style='o')
print('pearsons correlation: ',data['precipitation'].corr(data['temp_max']))
print('T test and P values: ',stats.ttest_ind(data['precipitation'],data['temp_max']))
```

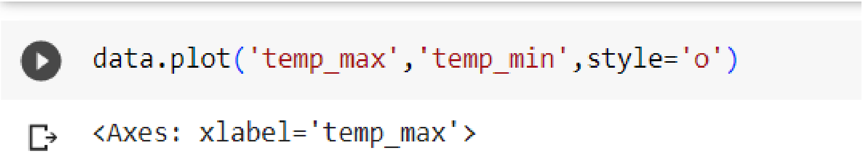
```
pearsons correlation:  -0.22855481643297046
T test and P values:  Ttest_indResult(statistic=-51.60685279531918, pvalue=0.0)
```





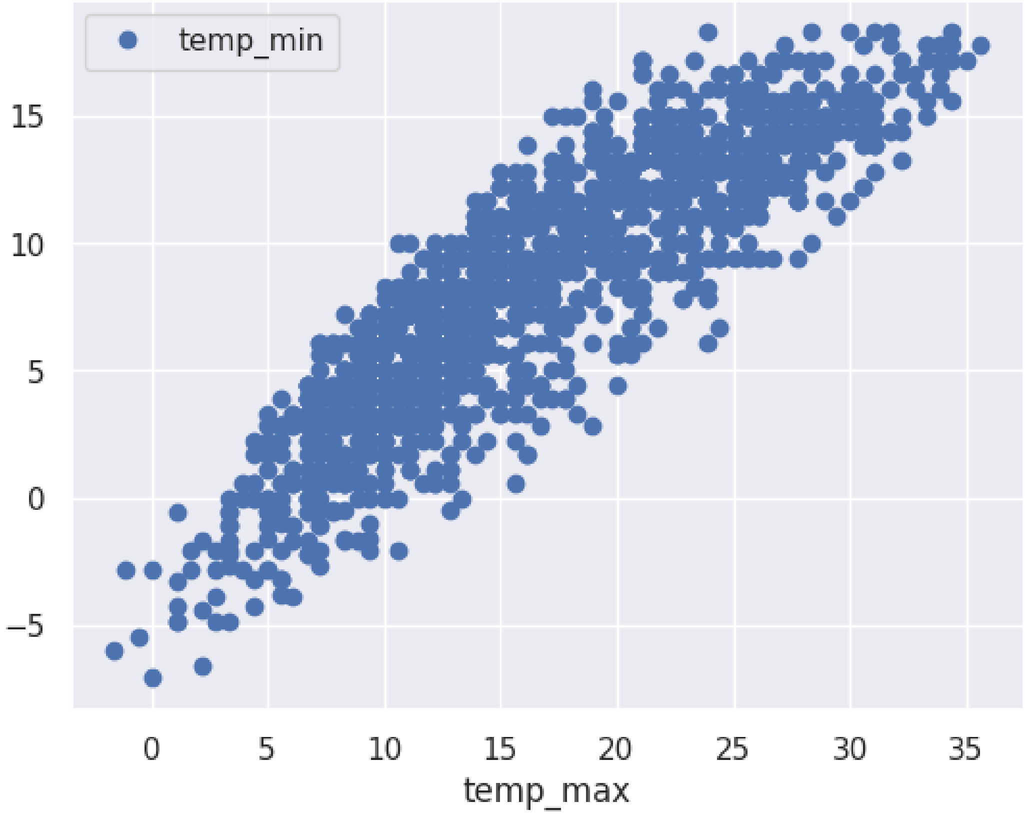
```
[ ] data.plot("wind",'temp_max',style='o')
    print('pearsons correlation: ',data['wind'].corr(data['temp_max']))
    print('T test and P values: ',stats.ttest_ind(data['wind'],data['temp_max']))

    pearsons correlation:  -0.16485663487495486
    T test and P values:  Ttest_indResult(statistic=-67.3601643301846, pvalue=0.0)
```

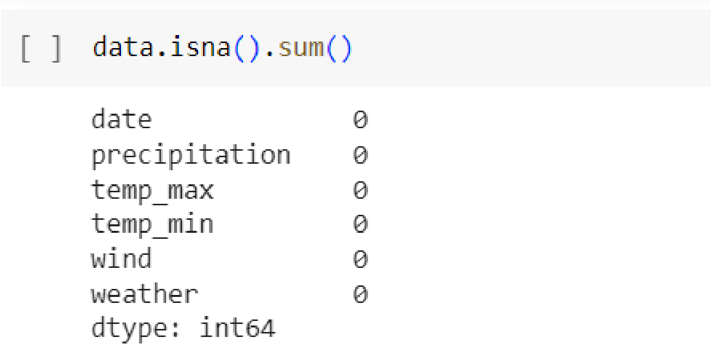




```
data.plot('temp_max','temp_min',style='o')
```

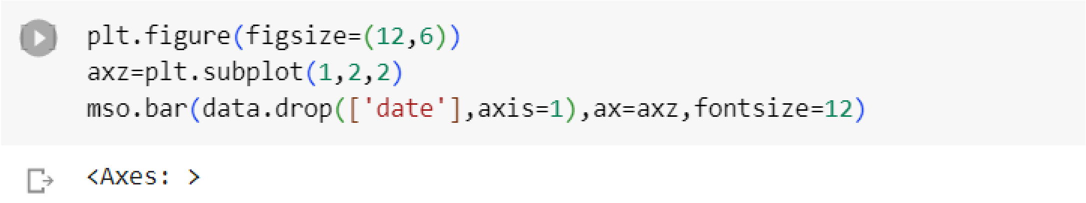
```
<Axes: xlabel='temp_max'>
```





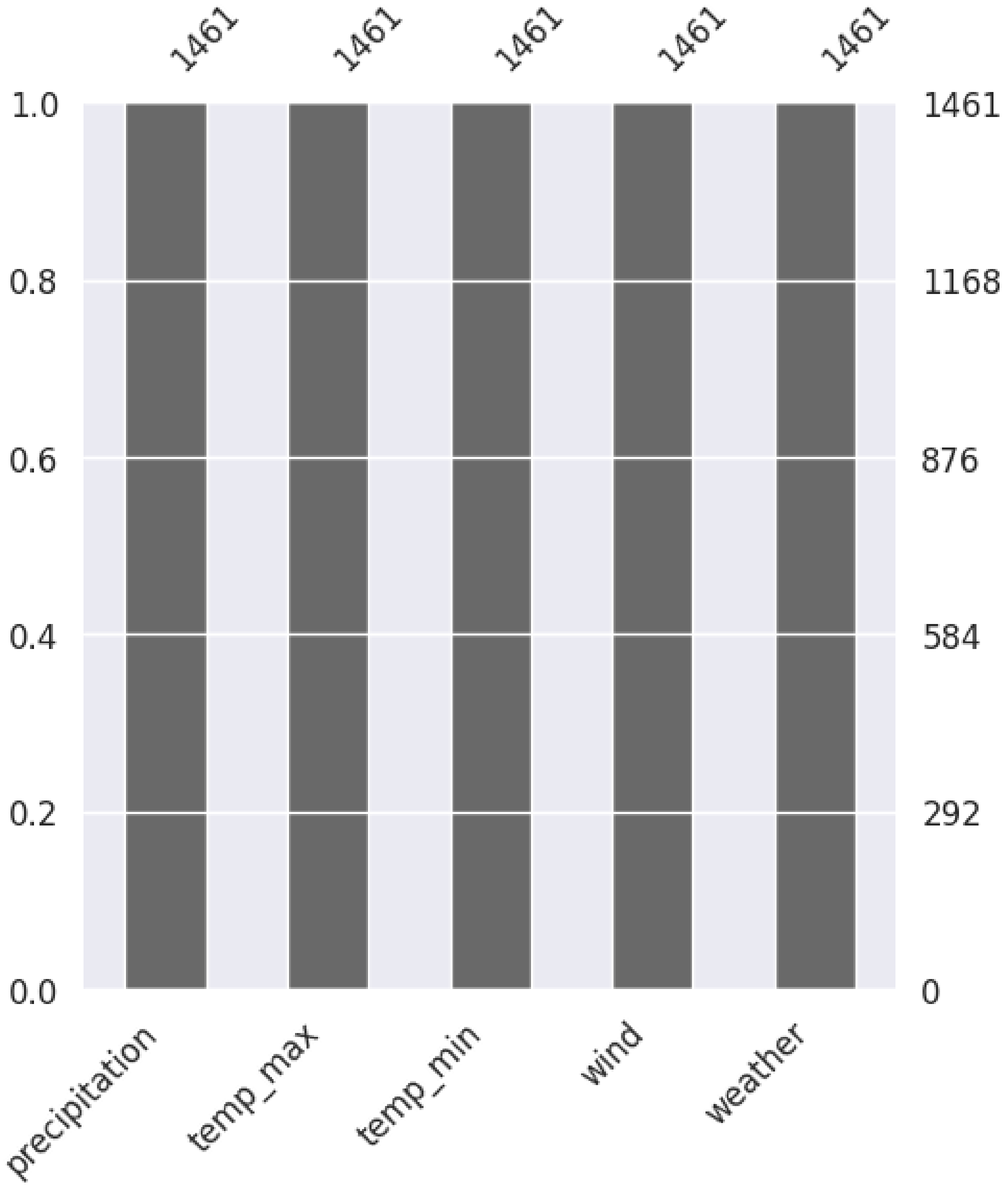
```
[ ]  data.isna().sum()
```

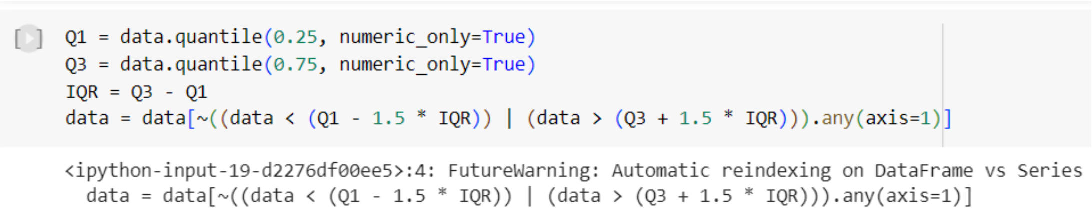
```
date             0
precipitation    0
temp_max         0
temp_min         0
wind             0
weather          0
dtype: int64
```



```
plt.figure(figsize=(12,6))
axz=plt.subplot(1,2,2)
mso.bar(data.drop(['date'],axis=1),ax=axz,fontsize=12)
```

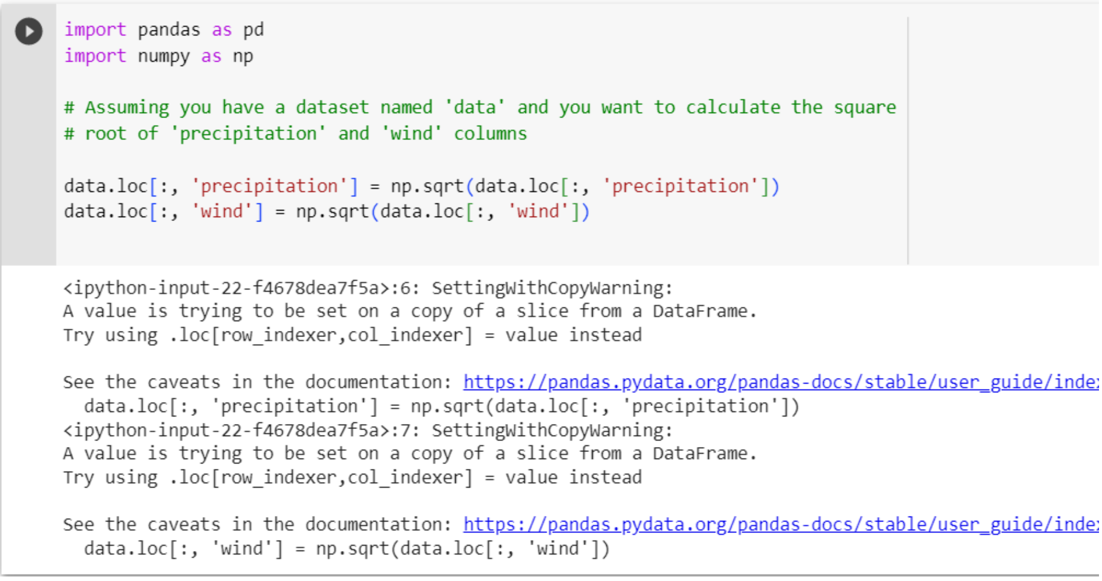
```
<Axes: >
```





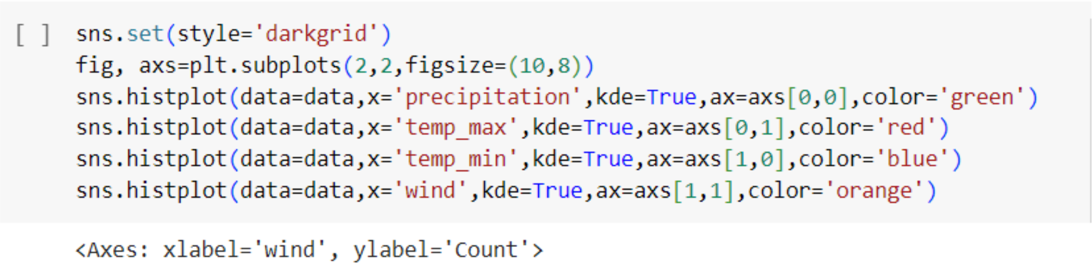
```
Q1 = data.quantile(0.25, numeric_only=True)
Q3 = data.quantile(0.75, numeric_only=True)
IQR = Q3 - Q1
data = data[~((data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
<ipython-input-19-d2276df00ee5>:4: FutureWarning: Automatic reindexing on DataFrame vs Series
  data = data[~((data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 * IQR))).any(axis=1)]
```



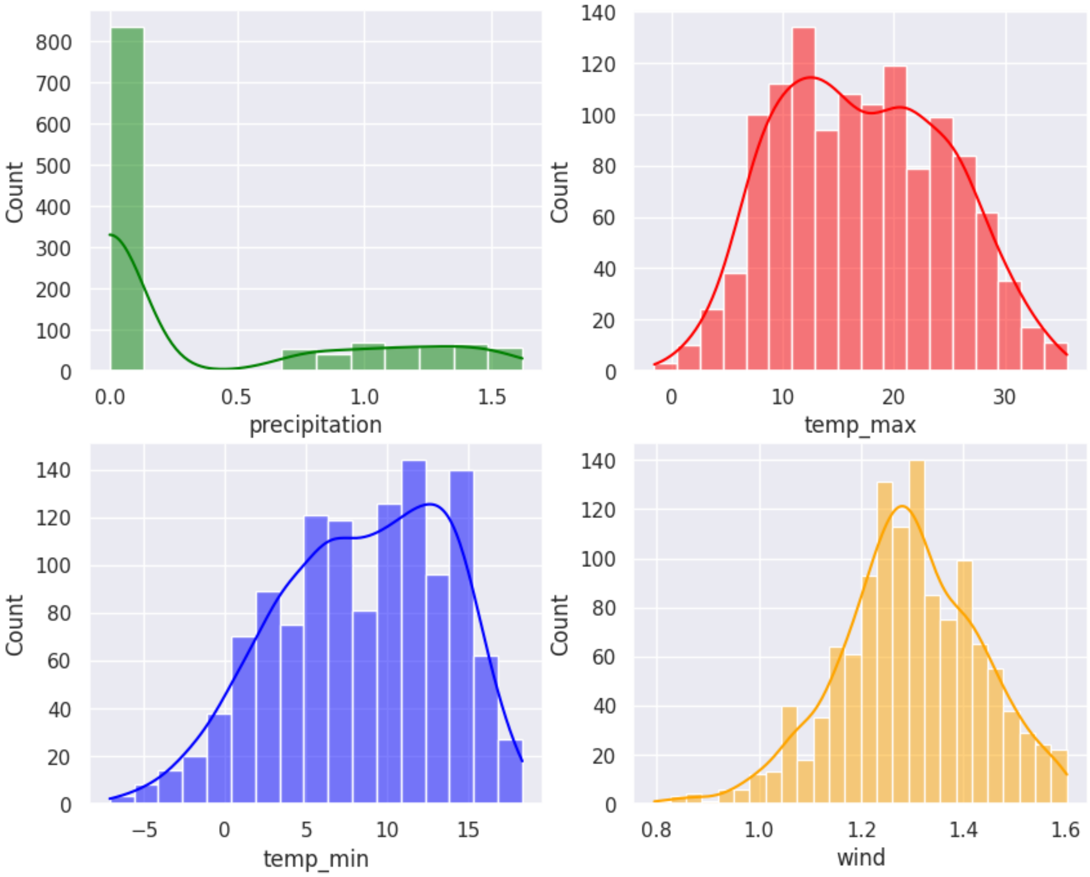
```
import pandas as pd
import numpy as np

# Assuming you have a dataset named 'data' and you want to calculate the square
# root of 'precipitation' and 'wind' columns

data.loc[:, 'precipitation'] = np.sqrt(data.loc[:, 'precipitation'])
data.loc[:, 'wind'] = np.sqrt(data.loc[:, 'wind'])
```

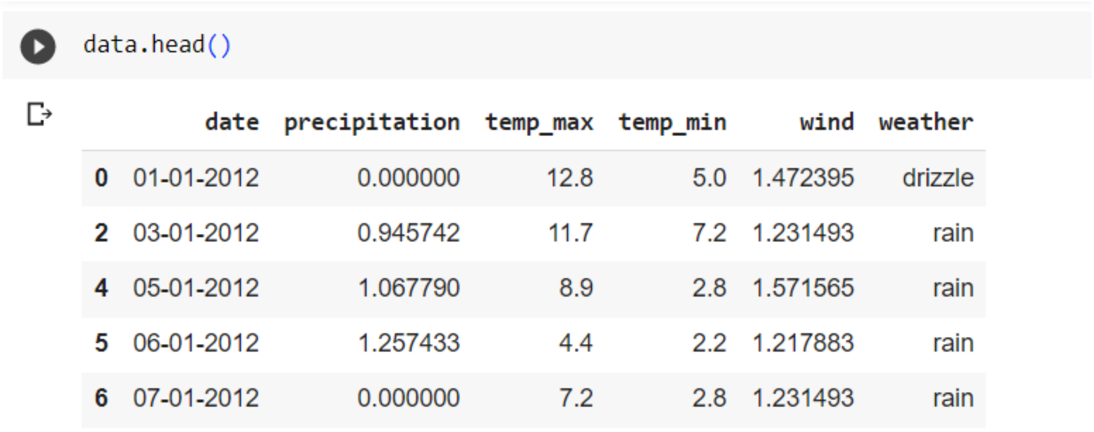
```
<ipython-input-22-f4678dea7f5a>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/inde
  data.loc[:, 'precipitation'] = np.sqrt(data.loc[:, 'precipitation'])
<ipython-input-22-f4678dea7f5a>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/inde
  data.loc[:, 'wind'] = np.sqrt(data.loc[:, 'wind'])
```



```
[ ]  sns.set(style='darkgrid')
     fig, axs=plt.subplots(2,2,figsize=(10,8))
     sns.histplot(data=data,x='precipitation',kde=True,ax=axs[0,0],color='green')
     sns.histplot(data=data,x='temp_max',kde=True,ax=axs[0,1],color='red')
     sns.histplot(data=data,x='temp_min',kde=True,ax=axs[1,0],color='blue')
     sns.histplot(data=data,x='wind',kde=True,ax=axs[1,1],color='orange')
```

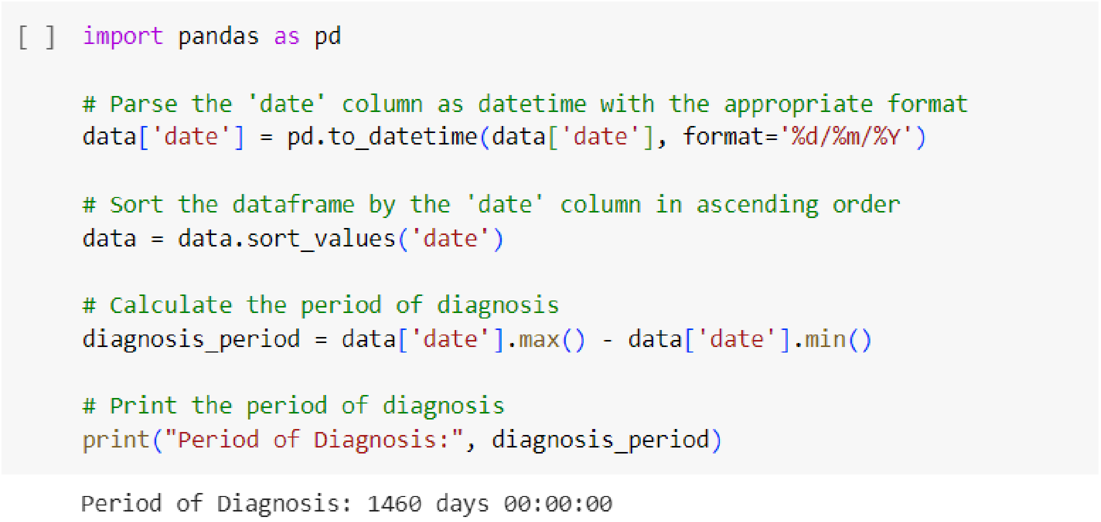
```
<Axes: xlabel='wind', ylabel='Count'>
```





```
data.head()
```

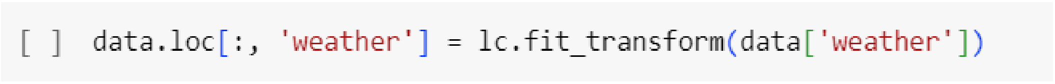
	date	precipitation	temp_max	temp_min	wind	weather
0	01-01-2012	0.000000	12.8	5.0	1.472395	drizzle
2	03-01-2012	0.945742	11.7	7.2	1.231493	rain
4	05-01-2012	1.067790	8.9	2.8	1.571565	rain
5	06-01-2012	1.257433	4.4	2.2	1.217883	rain
6	07-01-2012	0.000000	7.2	2.8	1.231493	rain



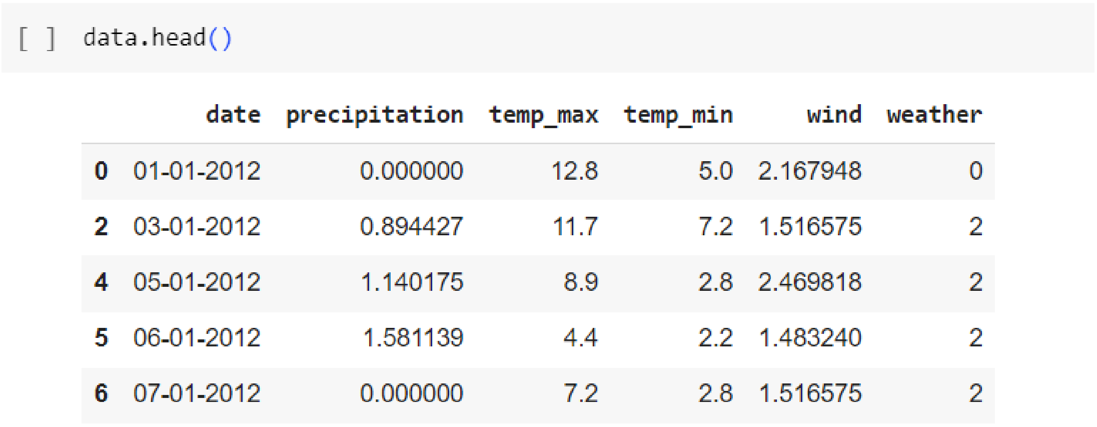
```
[ ] import pandas as pd

    # Parse the 'date' column as datetime with the appropriate format
    data['date'] = pd.to_datetime(data['date'], format='%d/%m/%Y')

    # Sort the dataframe by the 'date' column in ascending order
    data = data.sort_values('date')

    # Calculate the period of diagnosis
    diagnosis_period = data['date'].max() - data['date'].min()

    # Print the period of diagnosis
    print("Period of Diagnosis:", diagnosis_period)
```

```
Period of Diagnosis: 1460 days 00:00:00
```

4.2.2 DATA PRE-PROCESSING

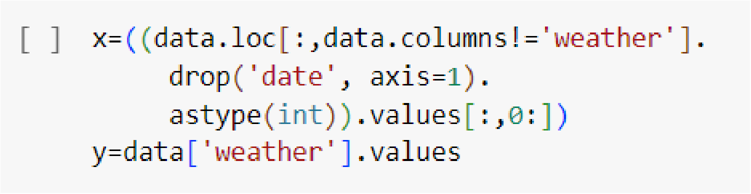


```
[ ]  data.loc[:, 'weather'] = lc.fit_transform(data['weather'])
```



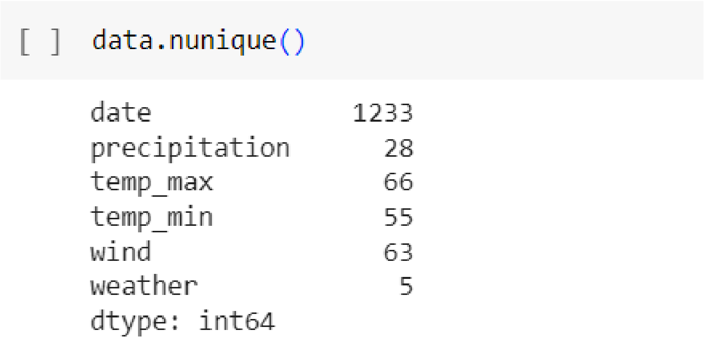
```
[ ] data.head()
```

	date	precipitation	temp_max	temp_min	wind	weather
0	01-01-2012	0.000000	12.8	5.0	2.167948	0
2	03-01-2012	0.894427	11.7	7.2	1.516575	2
4	05-01-2012	1.140175	8.9	2.8	2.469818	2
5	06-01-2012	1.581139	4.4	2.2	1.483240	2
6	07-01-2012	0.000000	7.2	2.8	1.516575	2



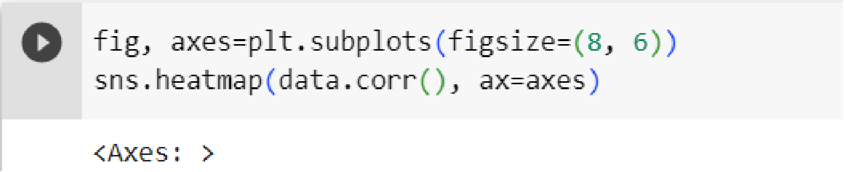
```
[ ]  x=((data.loc[:,data.columns!='weather'].
          drop('date', axis=1).
          astype(int)).values[:,0:])
     y=data['weather'].values
```

4.2.3 DATA VISUALIZATION



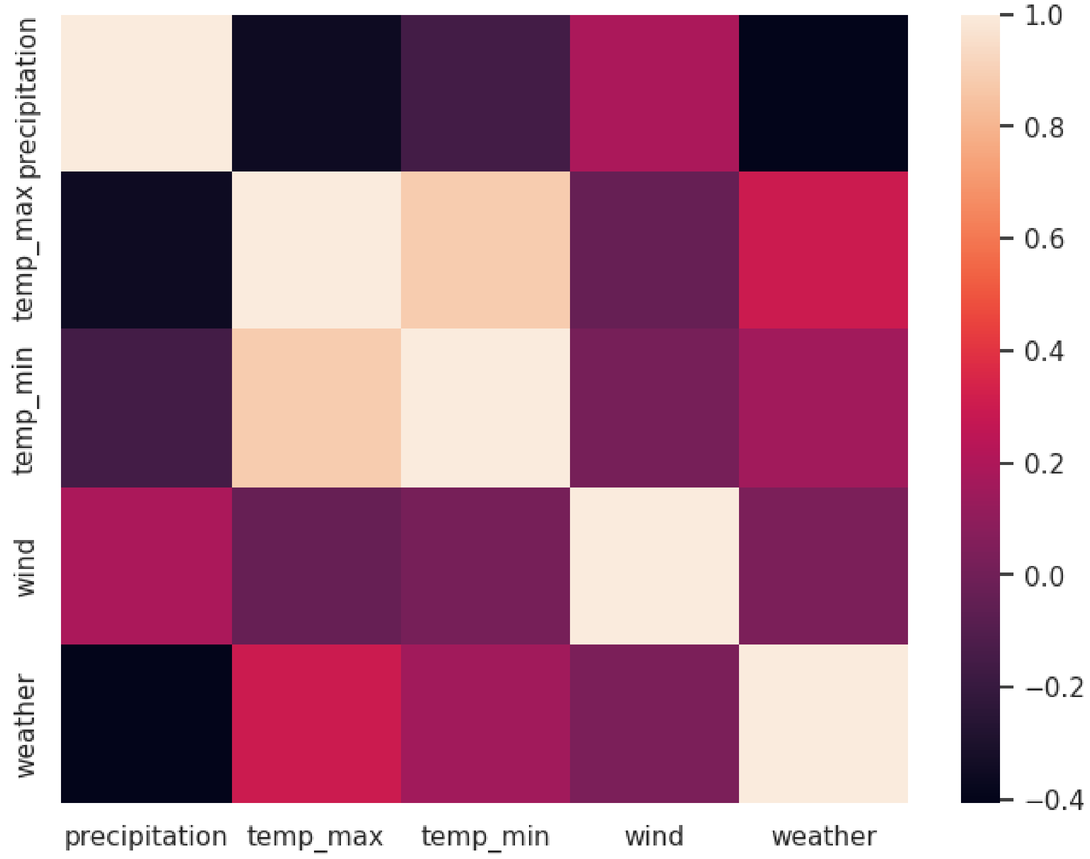
```
[ ]  data.nunique()
```

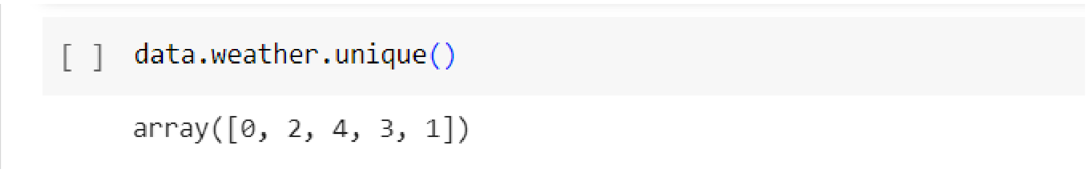
```
     date             1233
     precipitation      28
     temp_max           66
     temp_min           55
     wind               63
     weather             5
     dtype: int64
```



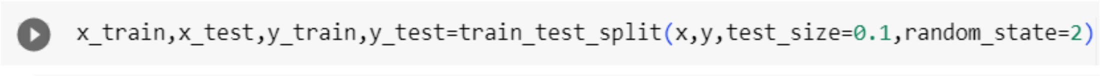
```
fig, axes=plt.subplots(figsize=(8, 6))
sns.heatmap(data.corr(), ax=axes)
```

```
<Axes: >
```





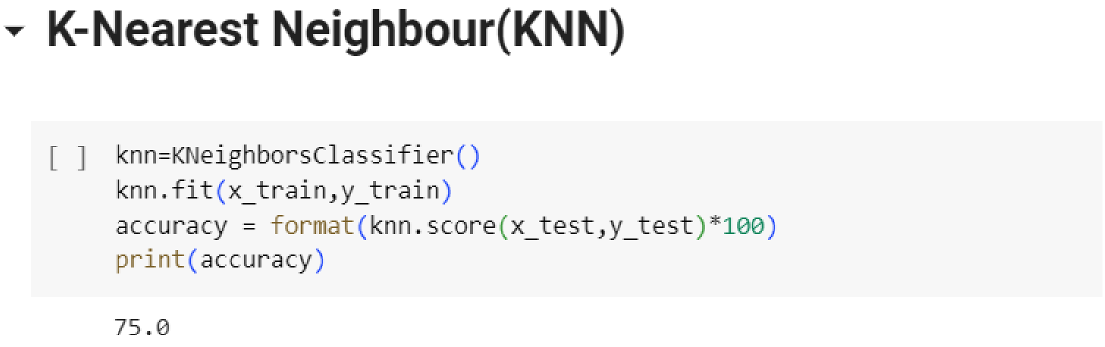
```
[ ]  data.weather.unique()

     array([0, 2, 4, 3, 1])
```



```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.1,random_state=2)
```

4.2.4 MACHINE LEARNING ALGORITHMS

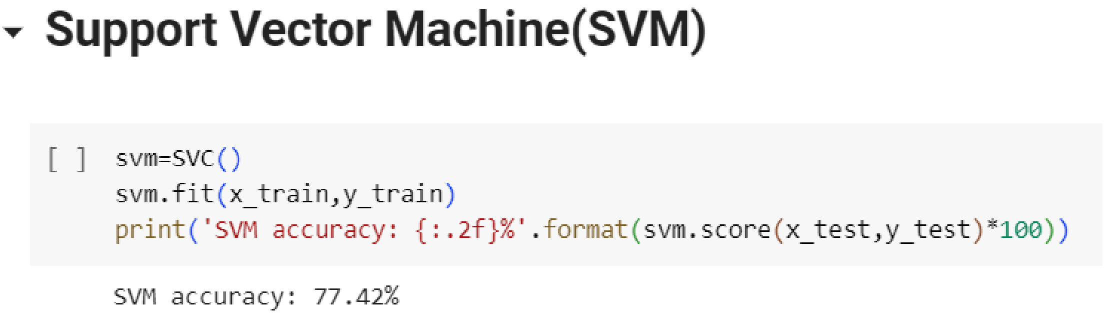
4.2.4.1 K-NEAREST NEIGHBOUR



K-Nearest Neighbour(KNN)

```
[ ]  knn=KNeighborsClassifier()
     knn.fit(x_train,y_train)
     accuracy = format(knn.score(x_test,y_test)*100)
     print(accuracy)

     75.0
```

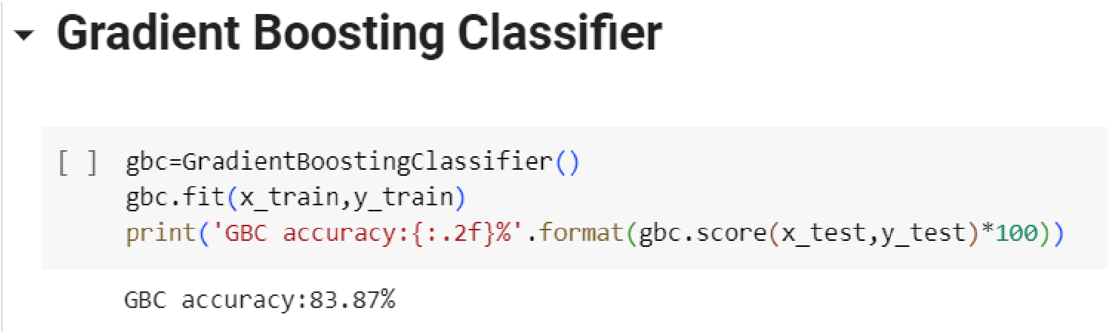
4.2.4.2 SUPPORT VECTOR MACHINE



Support Vector Machine(SVM)

```
[ ]  svm=SVC()
     svm.fit(x_train,y_train)
     print('SVM accuracy: {:.2f}%'.format(svm.score(x_test,y_test)*100))

     SVM accuracy: 77.42%
```

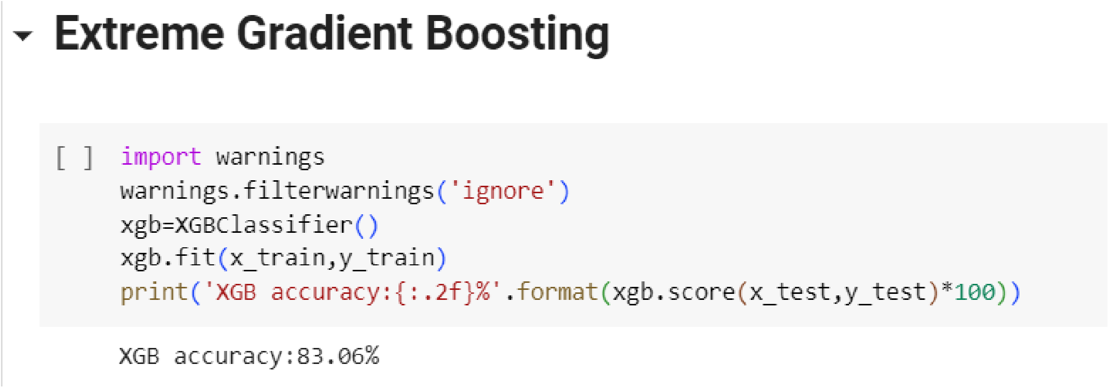
4.2.4.3 GRADIENT BOOSTING CLASSIFIER



Gradient Boosting Classifier

```
[ ]  gbc=GradientBoostingClassifier()
     gbc.fit(x_train,y_train)
     print('GBC accuracy:{:.2f}%'.format(gbc.score(x_test,y_test)*100))

     GBC accuracy:83.87%
```

4.2.4.4 EXTREME GRADIENT BOOSTING

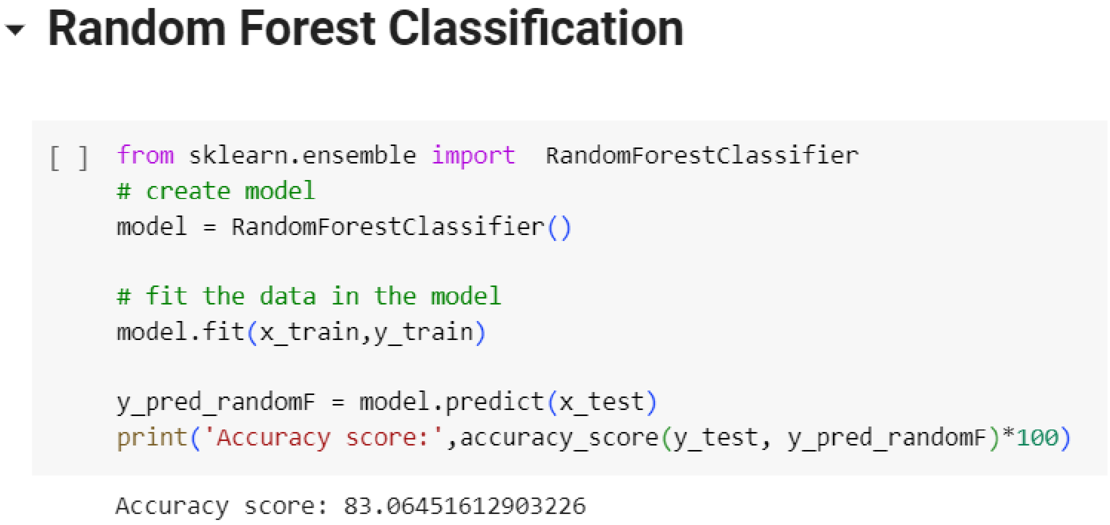


Extreme Gradient Boosting

```
[ ]  import warnings
     warnings.filterwarnings('ignore')
     xgb=XGBClassifier()
     xgb.fit(x_train,y_train)
     print('XGB accuracy:{:.2f}%'.format(xgb.score(x_test,y_test)*100))
```

```
XGB accuracy:83.06%
```

4.2.4.5 RANDOM FOREST CLASSIFICATION

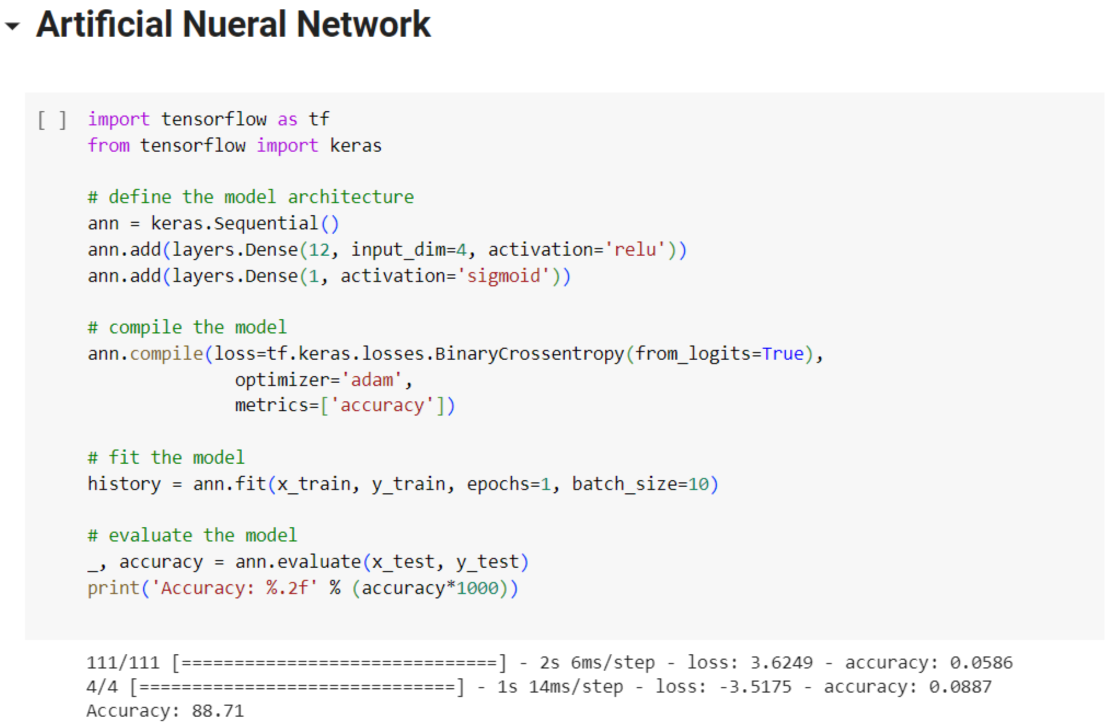


Random Forest Classification

```
[ ]  from sklearn.ensemble import  RandomForestClassifier
     # create model
     model = RandomForestClassifier()

     # fit the data in the model
     model.fit(x_train,y_train)

     y_pred_randomF = model.predict(x_test)
     print('Accuracy score:',accuracy_score(y_test, y_pred_randomF)*100)
```

```
Accuracy score: 83.06451612903226
```

4.2.4.6 ARTIFICIAL NUERAL NETWORK

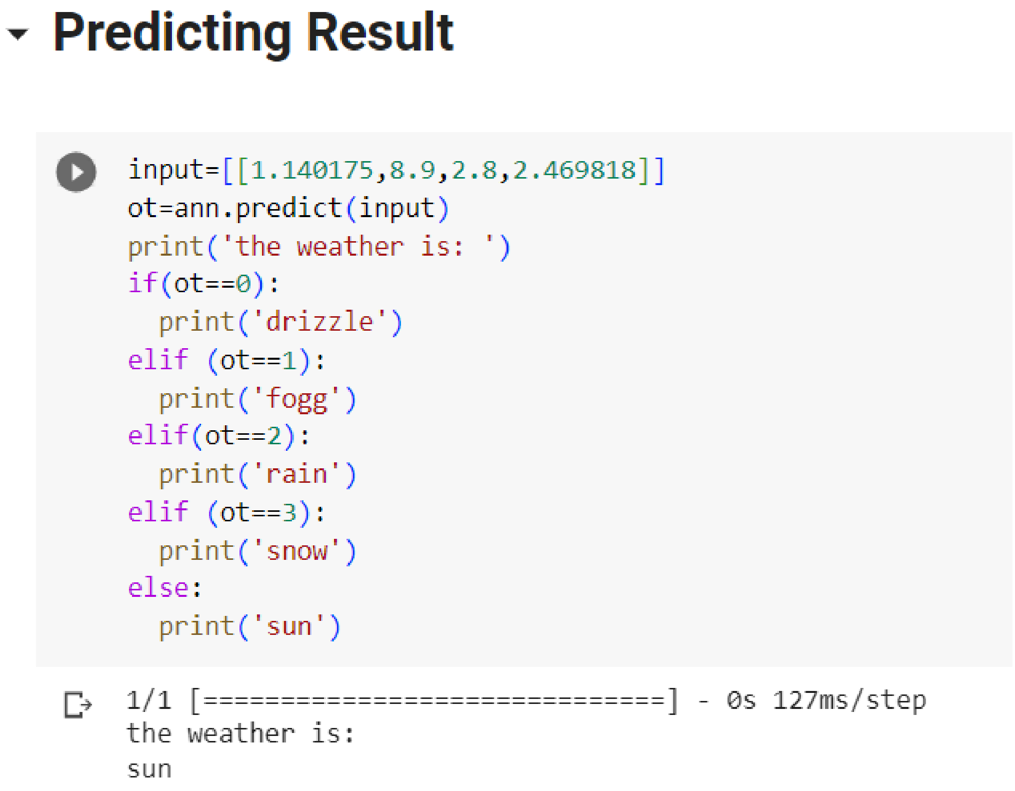


Artificial Nueral Network

```
[ ] import tensorflow as tf
    from tensorflow import keras

    # define the model architecture
    ann = keras.Sequential()
    ann.add(layers.Dense(12, input_dim=4, activation='relu'))
    ann.add(layers.Dense(1, activation='sigmoid'))

    # compile the model
    ann.compile(loss=tf.keras.losses.BinaryCrossentropy(from_logits=True),
                  optimizer='adam',
                  metrics=['accuracy'])

    # fit the model
    history = ann.fit(x_train, y_train, epochs=1, batch_size=10)

    # evaluate the model
    _, accuracy = ann.evaluate(x_test, y_test)
    print('Accuracy: %.2f' % (accuracy*1000))
```

```
111/111 [==============================] - 2s 6ms/step - loss: 3.6249 - accuracy: 0.0586
4/4 [==============================] - 1s 14ms/step - loss: -3.5175 - accuracy: 0.0887
Accuracy: 88.71
```

4.2.5 PREDICTED OUTPUT



Predicting Result

```
input=[[1.140175,8.9,2.8,2.469818]]
ot=ann.predict(input)
print('the weather is: ')
if(ot==0):
  print('drizzle')
elif (ot==1):
  print('fogg')
elif(ot==2):
  print('rain')
elif (ot==3):
  print('snow')
else:
  print('sun')
```

```
1/1 [==============================] - 0s 127ms/step
the weather is:
sun
```

4.3 EXPERIMENTAL ANALYSIS / TESTING

4.3.1 DATASET

The dataset, which is considered from Koggle website, consists of all dataset in online with free downloads. The dataset is names as "seattle-weather". It consists of different types of all weather conditions. It contains of Date, precipitation, maximum temperature which is named there as 'temp_max', minimum temperature which is also changed the attribute name in dataset is 'temp_min', wind and weather. The weather attribute contains different types of entries called as drizzle, rain, sun, snow, and fogg. For each attribute contain different readings and different entries. Along with dataset and their attributes, 'weather' attribute is considered as the target label or output label. This dataset is considered to compare the accuracy results obtained and checking the best accuracy model. This will help us for future comparison too.

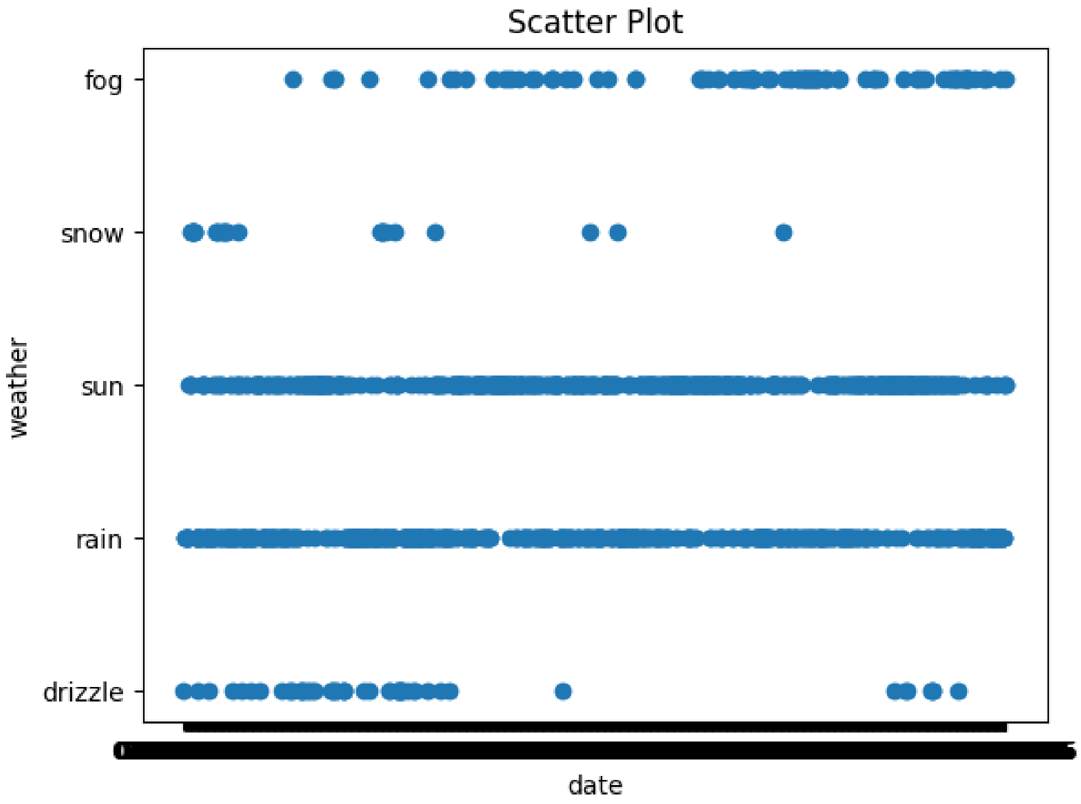
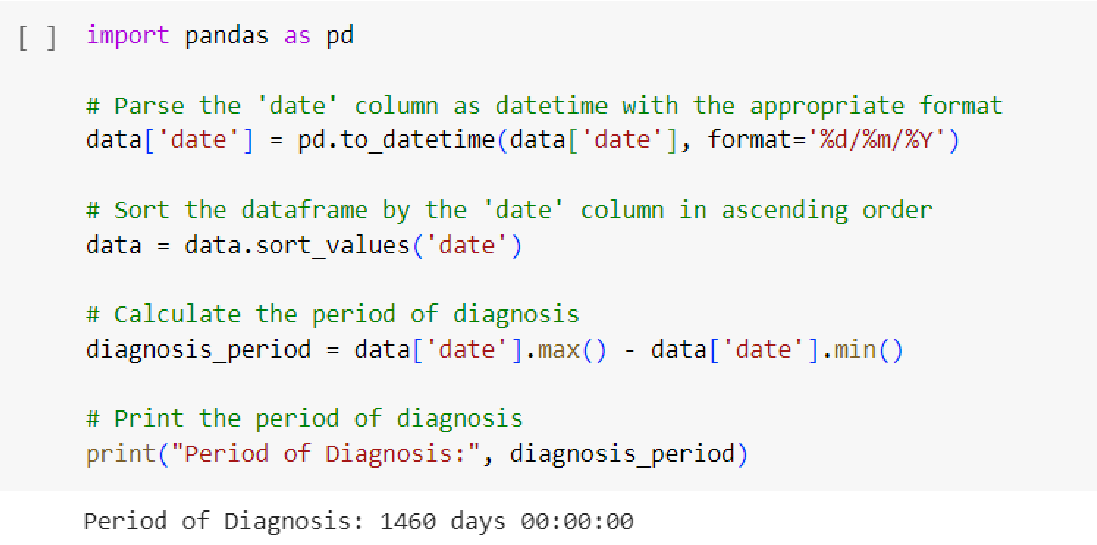


Fig. 6 Scatter plot of date and weather attribute of the dataset

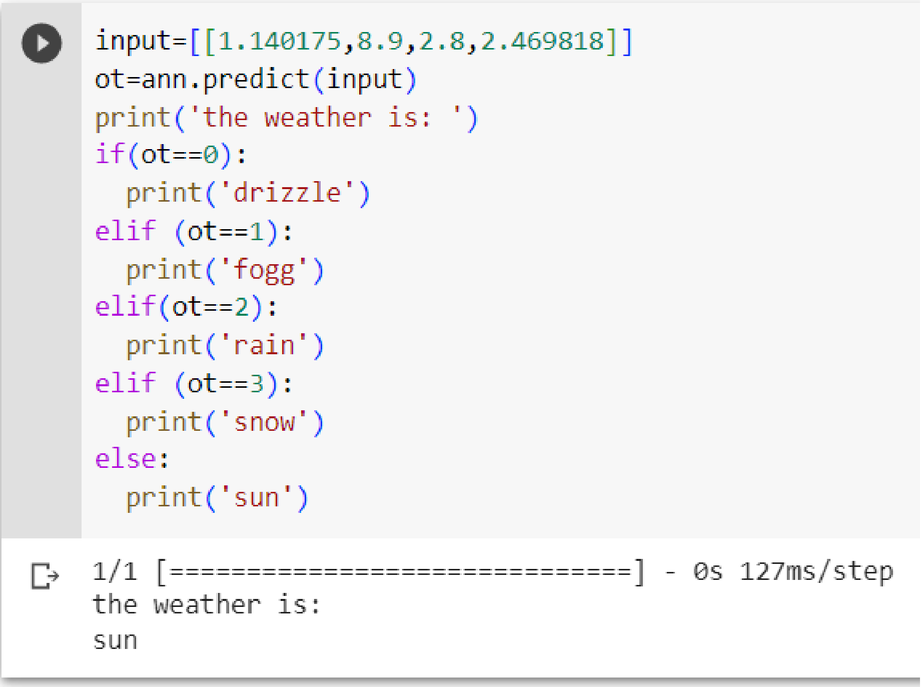
4.3.2 ANALYSIS

The period of diagnosis and the final output of dataset obtained for the input data is given in a text format as shown in the figure below.



```
[ ]  import pandas as pd

     # Parse the 'date' column as datetime with the appropriate format
     data['date'] = pd.to_datetime(data['date'], format='%d/%m/%Y')

     # Sort the dataframe by the 'date' column in ascending order
     data = data.sort_values('date')

     # Calculate the period of diagnosis
     diagnosis_period = data['date'].max() - data['date'].min()

     # Print the period of diagnosis
     print("Period of Diagnosis:", diagnosis_period)

     Period of Diagnosis: 1460 days 00:00:00
```

Fig. 7 Period of diagnosis of dataset



```
input=[[1.140175,8.9,2.8,2.469818]]
ot=ann.predict(input)
print('the weather is: ')
if(ot==0):
  print('drizzle')
elif (ot==1):
  print('fogg')
elif(ot==2):
  print('rain')
elif (ot==3):
  print('snow')
else:
  print('sun')
```

```
1/1 [==============================] - 0s 127ms/step
the weather is:
sun
```

Fig.8 Final output of dataset in the form of text

4.4 RESULTS

After uploading dataset and implementing some basic functions, the dataset has to be pre processed for raw dataset which suits for all algorithms without any modifications further. After data pre- processing, data visualization is done to get the data representation o raw dataset in the form of HeatMap correlation. And after all this process, some supervised machine learning techniques has been applied named as K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Gradient Boosting Classifier (GBC), Extreme Gradient Boosting Classification (XGB), Random Forest Classification (RFC) and Artificial Neural Network (ANN). Out of these algorithms ANN gave the best accuracy rate of 88.71. So, for prediction any random data which corresponds to dataset has been used to predict the future Weather condition with ANN model as mentioned previous.

Algorithms	Accuracy Rate
K-Nearest Neighbour	75.0
Support Vector Machine	77.42
Gradient Boosting Classification	83.87
Extreme Gradient Boosting Classification	83.06
Random Forest Classification	83.06
Artificial Neural Networks	88.71

Table. 1 Accuracy Rate of Machine Learning Algorithms

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

Weather prediction is a complex task that involves analysing various meteorological factors and historical data to forecast future weather conditions. Machine learning and artificial intelligence techniques have proven to be valuable in weather prediction, offering improved accuracy and efficiency compared to traditional methods.

Machine learning algorithms, such as K-Nearest Neighbours, Random Forests Classification, Support Vector Machine, Gradient Boosting Classifier, Extreme Gradient Boosting Classifier and Artificial Neural Network have been successfully applied to weather prediction. These algorithms can analyse large volumes of data, identify patterns, and make predictions based on historical weather patterns and meteorological variables.

The integration of machine learning techniques, advanced data analysis, and continuous model improvement has greatly advanced weather prediction capabilities. These advancements contribute to more accurate forecasts, improved disaster preparedness, and better decision-making in various sectors that rely on weather information, such as agriculture, transportation, energy, and emergency management.

In conclusion, out of those supervised machine learning algorithms, ANN gave the best accuracy results to predict the accuracy of environment. So, here ANN has been taken for predicting the weather condition. Here, for this, we use random data which corresponds to the dataset. Based on the random dataset only it is going to predict the future time weather condition state.

REFERENCES

1. Weather Forecasting using Satellite Image Processing and Artificial Neural Networks by Nilay S. Kapadia, Urmil Parikh in IJCSIS vol 14, No.11, Nov 2016.

2. https://github.com/MArya80/Australia-Weather-Prediction

3. Machine Learning Applied to Weather Forecasting Mark Holmstrom, Dylan Liu, Christopher Vo Stanford University, December 15, 2016.

4. [3] ANALYSIS ON THE WEATHER FORECASTING AND TECHNIQUES Janani.B, Priyanka Sebastian, Jan 2014.

5. https://github.com/NAMYUNWOO/Weather_prediction_GPR

6. Suvendra Kumar Jayasingh, Jibendu Kumar Mantri, Sipali Pradhan, “Smart Weather Prediction Using Machine Learning”, May 2022. DOI:10.1007/978-981-19-0901-6_50

7. https://towardsdatascience.com/weather-forecasting-with-machine-learning-using-python-55e90c346647

8. A H M Jakaria, Md Mosharaf Hossain, Mohammad Ashiqur Rahman, “Smart Weather Forecasting Using Machine Learning: A Case Study in Tennessee”, August 2020.

9. “Weather Forecast Prediction: An Integrated Approach for Analyzing and Measuring Weather Data” by Munmun Biswas, Tanni, Sayantanu Barua in the year 2018.

10. https://github.com/neetika6/Machine-Learning-Model-for-Weather-Forecasting