

DATA SCIENCE WITH AI & ML

Submitted by : Submitted to:

1.Divyansh Singh Tanwer ->Rishu Trivedi

2.Vaibhav Singh Bains LNB mentor

3.Shivam Purohit

What is NumPy?

NumPy is a Python library used for working with arrays.

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

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

NumPy stands for Numerical Python.

Why Use NumPy?

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

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

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

Arrays are very frequently used in data science, where speed and resources are very important.

import numpy as np

# Creating array from list with type float

a = np.array([[1, 2, 4], [5, 8, 7]], dtype = 'float')

print ("Array created using passed list:\n", a)

# Creating array from tuple

b = np.array((1 , 3, 2))

print ("\nArray created using passed tuple:\n", b)

What is Pandas?

Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Why Use Pandas?

Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science.

import pandas as pd

import numpy as np

# Creating empty series

ser = pd.Series()

print("Pandas Series: ", ser)

# simple array

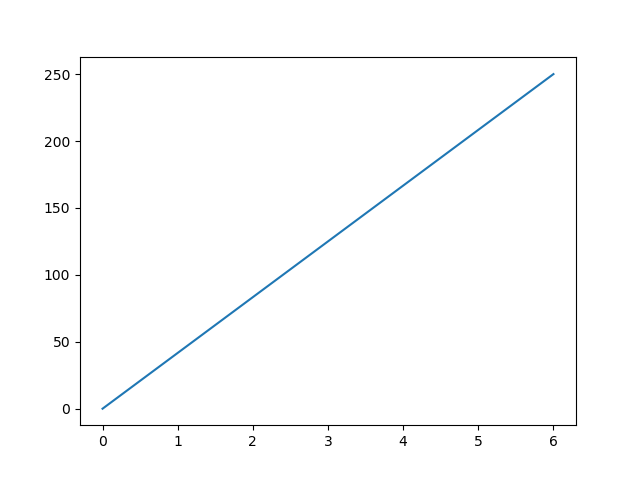
data = np.array(['a', 'b', 'c', 'd', 'e'])

ser = pd.Series(data)

print("Pandas Series:\n", ser)

**Matplotlib Pyplot**

import matplotlib.pyplot as plt  
import numpy as np  
  
xpoints = np.array([0, 6])  
ypoints = np.array([0, 250])  
  
plt.plot(xpoints, ypoints)  
plt.show()



## Project: Demand Forecasting Industry: E-Commerce, Retail

The "Retail - Demand Forecasting" project aims to develop an advanced demand forecasting model to optimize retail inventory management and supply chain operations. By analyzing sales history, promotional events, seasonal trends, and economic indicators, the project's objective is to accurately predict product demand and minimize issues such as excess inventory and stockouts. The ultimate goal is to enhance inventory planning, reduce costs, and improve overall retail efficiency

import pandas as pd

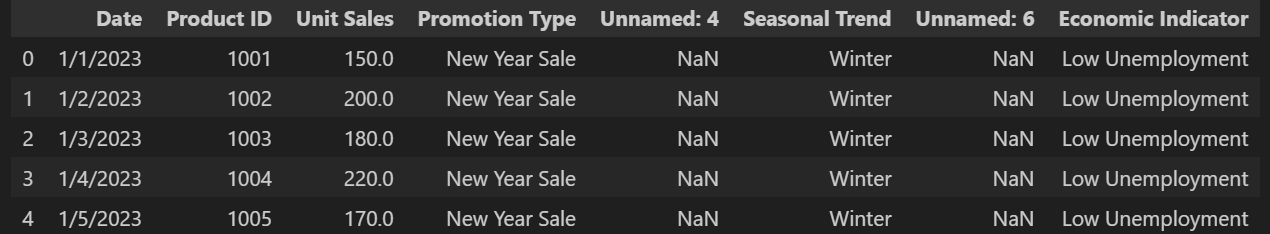
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

data = pd.read\_csv(r"C:\Users\vaibh\Downloads\data.csv.csv")

data.head()

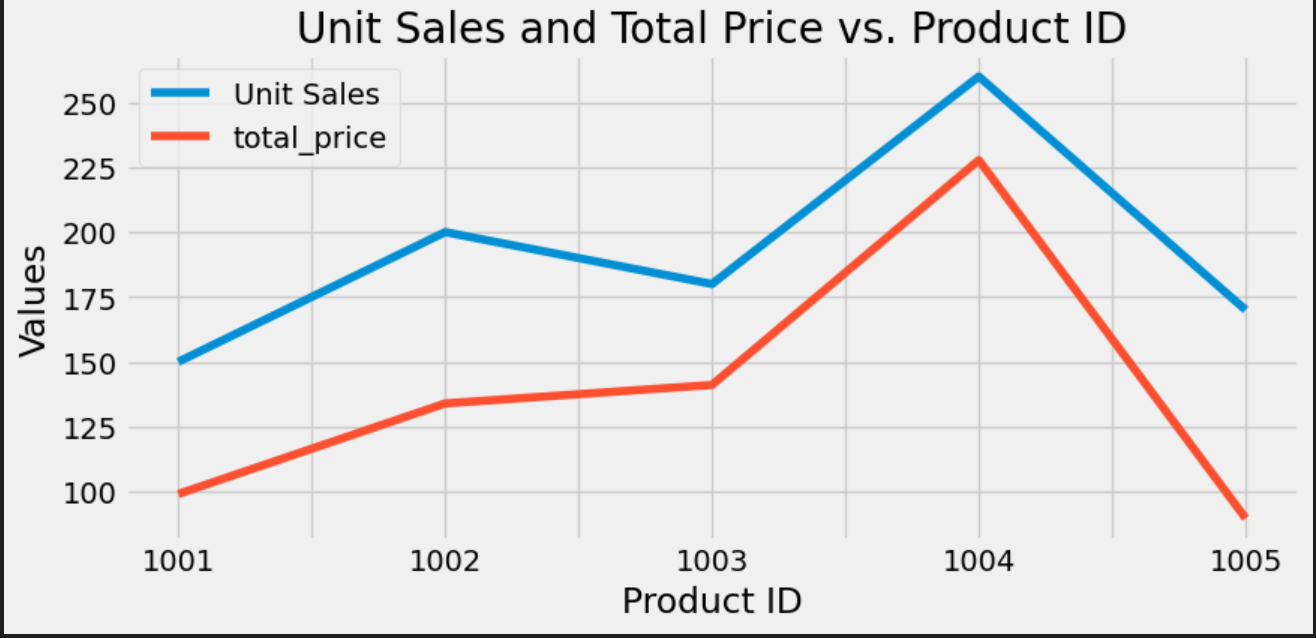


## Exploratory Analysis

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.

For example: from the project



## Types od EDA

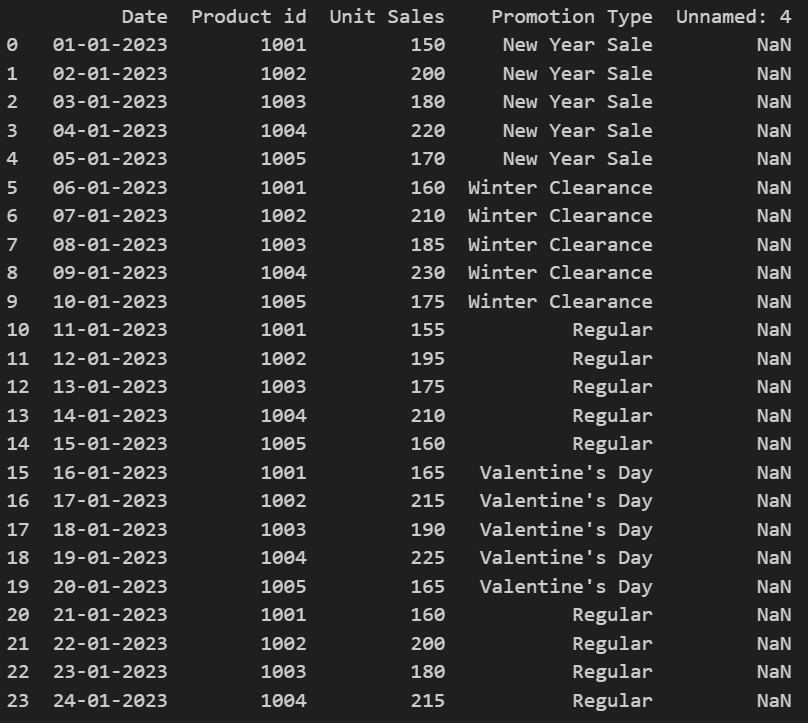
**1.Univariate Analysis**

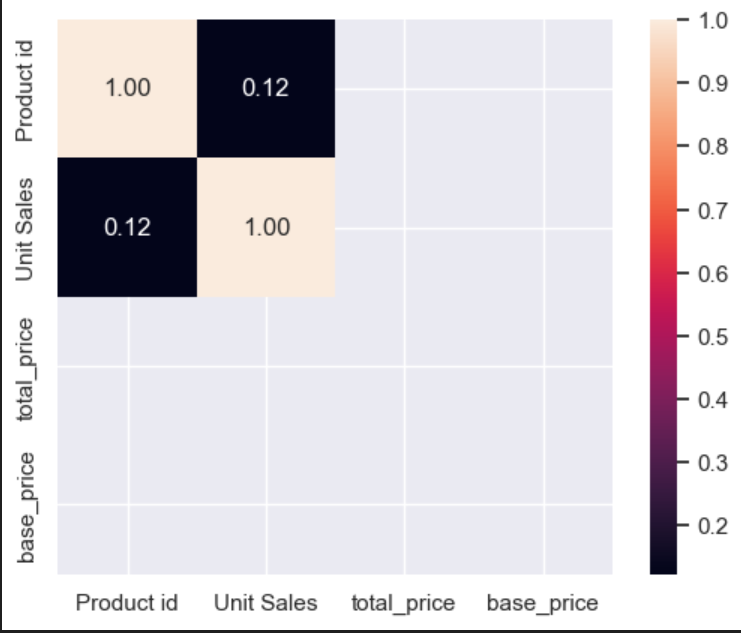
**2. Bivariate Analysis**

## Feature Engineering

## Feature engineering is the process of **transforming raw data into features that are suitable for machine learning models**. In other words, it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning models.

The success of machine learning models heavily depends on the quality of the features used to train them. Feature engineering involves a set of techniques that enable us to create new features by combining or transforming the existing ones. These techniques help to highlight the most important patterns and relationships in the data, which in turn helps the machine learning model to learn from the data more effectively.

For Example: from the project



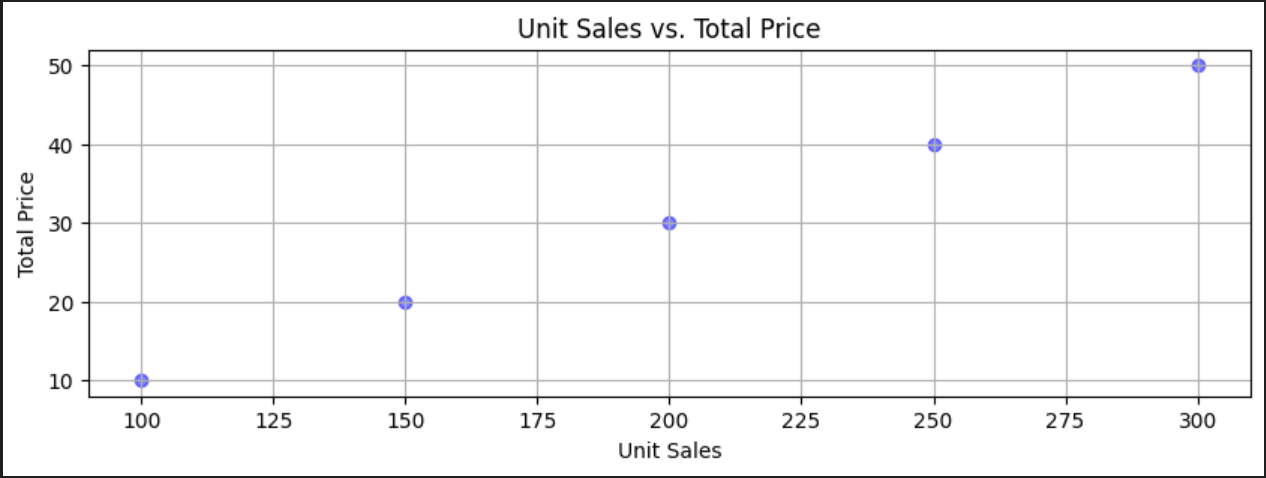
## Random Forest Regression

## What is Random Forest Regression?

Random Forest is an [ensemble technique](https://www.geeksforgeeks.org/ensemble-methods-in-python/) capable of performing both [regression and classification](https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/) tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as [bagging](https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/). The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

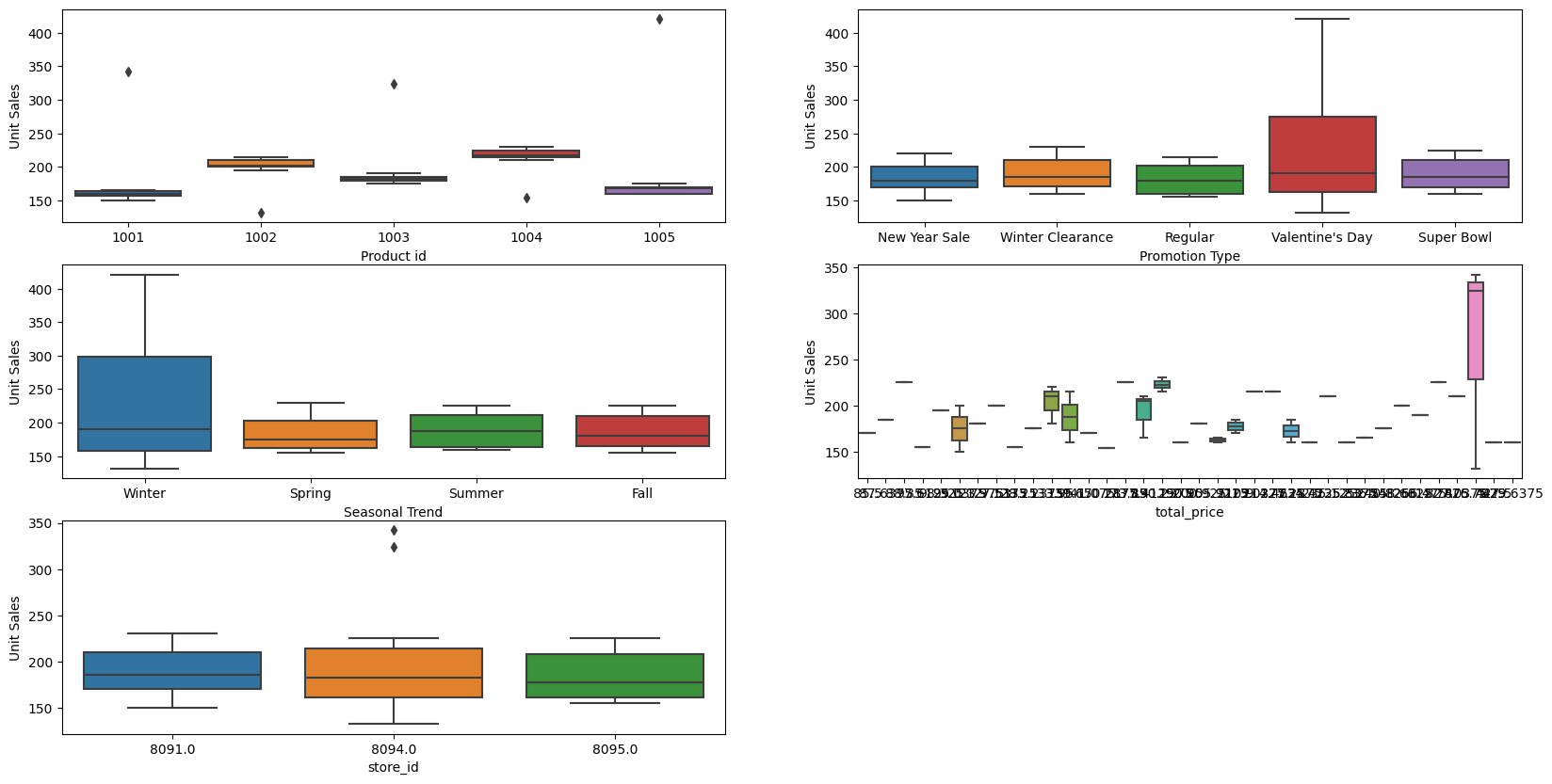
Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

For example: from the project

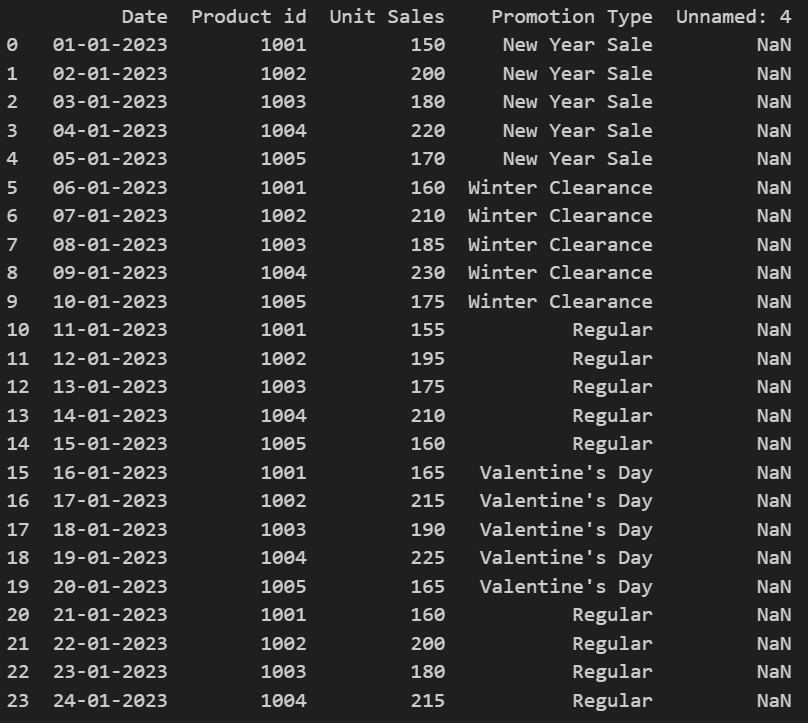


Linear Regression

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression. The goal of the algorithm is to find the best linear equation that can predict the value of the dependent variable based on the independent variables. The equation provides a straight line that represents the relationship between the dependent and independent variables.



Visualisation Of Data

*Data visualization is the graphical representation of information and data in a pictorial or graphical format(Example: charts, graphs, and maps). Data visualization tools provide an accessible way to see and understand trends, patterns in data, and outliers. Data visualization tools and technologies are essential to analyzing massive amounts of information and making data-driven decisions.*

import pandas as pd

# Sample time-series data (replace with your own dataset)

data = pd.read\_csv(r"C:\Users\vaibh\Downloads\demand data 2.csv")

df = pd.DataFrame(data)

# Convert the timestamp column to datetime

df['instant'] = pd.to\_datetime(df['instant'])

# Extract temporal features

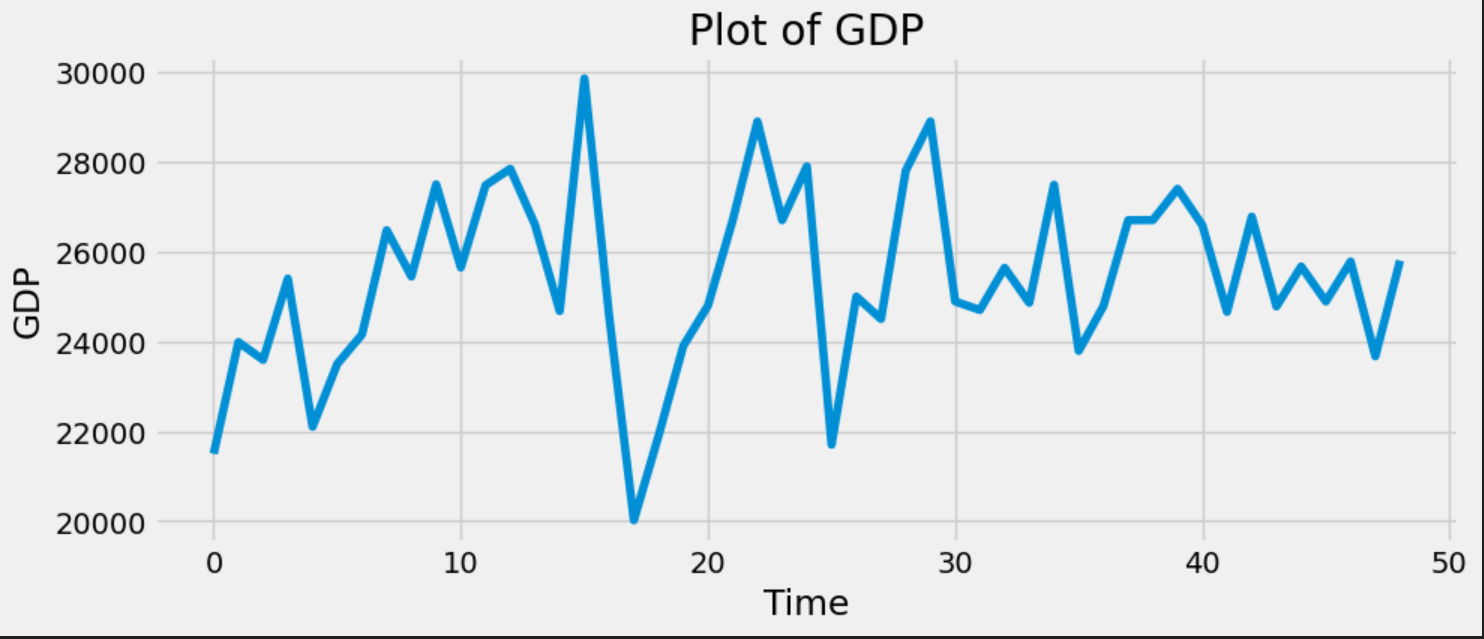
df['day'] = df['instant'].dt.day  # 0 for Monday, 6 for Sunday

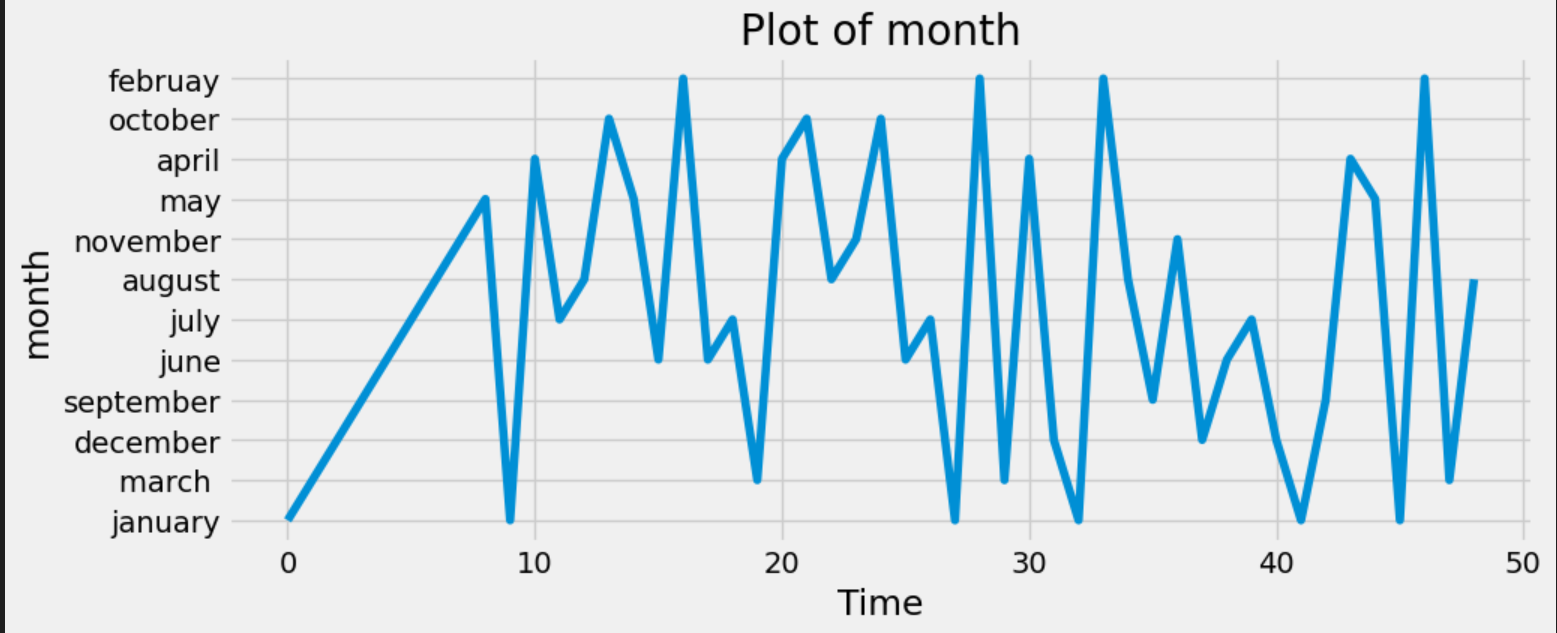
df['month'] = df['instant'].dt.month  # 1 to 12

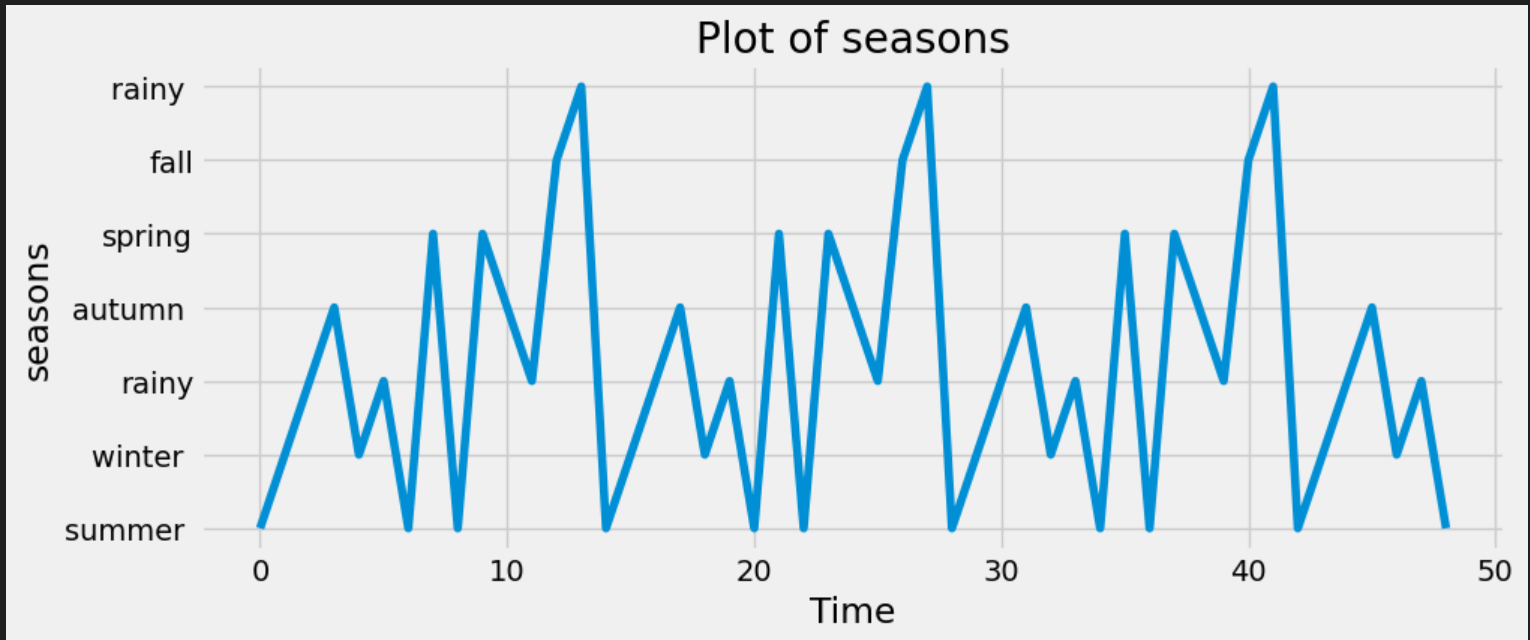
df['year'] = df['instant'].dt.year

# Print the updated DataFrame

print(df)







## MAE and RMSE

## The MAE is a linear score which means that all the individual differences are weighted equally in the average. The RMSE is a quadratic scoring rule which measures the average magnitude of the error. The equation for the RMSE is given in both of the references.

