COVID 19 Case Analysis Using Data Analytics Tool

DATE	**31-10-2023**
TEAM ID	**3925**
PROJECT NAME	**COVID 19 CASE ANYSIS**

TABLE OF CONTENTS

1.Introduction :

The COVID-19 Case Analysis Data Analytics Project is a comprehensive undertaking aimed at leveraging IBM Cognos and advanced data visualization techniques to gain actionable insights from the vast and dynamic dataset related to the COVID-19 pandemic. The project's primary objectives are to monitor, analyze, and visualize COVID-19 data to support informed decision-making, enhance public health response, and provide valuable information to various stakeholders.

The project gathers data from diverse sources, including national and local health departments, global health organizations, and open data repositories, to ensure comprehensive coverage of COVID-19 cases, mortality, testing, vaccination rates, and related information.

2. Problem Statement:

Designing a project to analyze COVID-19 cases and deaths using IBM cognos, The objective is to compare and contrast the mean and standard deviation of cases and deaths, which is a valuable undertaking. This project will involve data analysis, visualization, and deriving insights from the data.

3. Tools and Libraries:

Python:

✓ Python is a versatile, high-level programming language known for its simplicity and readability. It's widely used in data analysis, machine learning, and scientific computing due to its extensive libraries and frameworks.

Pandas:

✓ Pandas is a Python library for data manipulation and analysis. It provides data structures like DataFrames and Series, making it easy to work with structured data. Pandas is essential for data loading, cleaning, and transformation.

NumPy:

✓ NumPy is another Python library that focuses on numerical computing. It provides support for large, multi-dimensional arrays and matrices, as well as a variety of mathematical functions to operate on these arrays. It's fundamental for numerical data processing.

Matplotlib:

✓ Matplotlib is a popular data visualization library in Python. It allows you to create static, animated, or interactive visualizations in a wide range of formats, including line plots, bar charts, scatter plots, and more. It's excellent for visualizing data and trends.

Seaborn:

✓ Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for creating attractive and informative statistical graphics. Seaborn simplifies the creation of complex visualizations and is often used for creating aesthetically pleasing charts.

Scikit-Learn:

✓ Scikit-Learn, also known as sklearn, is a powerful machine learning library in Python. It offers a wide range of tools for data preprocessing, model selection, training, and evaluation. It's especially useful for building predictive models and performing machine learning tasks.

In the context of our project, these tools and libraries play the following roles:

- Python serves as the programming language for the project, providing a flexible and accessible environment for data analysis and modeling.
- Pandas is used for data manipulation and analysis, including loading, cleaning, and organizing the covid19 case data.

- NumPy complements Pandas by providing fundamental support for numerical operations and handling multi-dimensional arrays, which are often used in data analysis.
- Matplotlib is employed for creating visualizations that help in understanding covid trends and conveying insights effectively.
- Seaborn enhances the visualization process by providing a higher-level interface to create aesthetically pleasing and informative statistical graphics.
- Scikit-Learn plays a crucial role in building the predictive model that estimates cases and deaths in future.

4. Model Training and Evaluation:

Support Vector Machine(SVM):

The accuracy of a Support Vector Machine (SVM) model for a COVID-19 dataset depends on various factors, including the specific dataset you're using, the features you've selected, the kernel type (linear, radial basis function, etc.), and the hyperparameters chosen for the SVM.

To determine the accuracy of SVM model, perform the following steps:

- **Data Splitting:** Split your dataset into a training set and a testing set. This allows you to train the model on one subset of the data and evaluate its accuracy on another.
- **Feature Selection/Engineering**: Carefully choose the relevant features , like deaths, cases, year, month, day, and possibly countries/territories. Make sure the features have a significant impact on predicting COVID-19 cases.
- **Preprocessing:** Preprocess the data, which includes handling missing values, scaling, encoding categorical variables such as countries/Teritorries, and dealing with outliers.
- **Model Selection**: Select the appropriate kernel type (e.g., linear, radial basis function) and hyperparameters for your SVM model.
- **Training:** Train the SVM model using your training data. Make sure to set the random seed or random state to ensure reproducibility.
- **Evaluation:** After training, use your testing dataset to evaluate the model's performance. Common evaluation metrics for regression tasks (like predicting COVID-19 cases) include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared (R^2).

5.Mean and Standard Deviation of Cases and Deaths:

```
mean_cases = covid_data['cases'].mean()
mean_deaths = covid_data['deaths'].mean()
print("The mean_cases =", mean_cases)
print('The mean_deaths =' , mean_deaths)
```

```
The mean_cases = 3661.010989010989
The mean_deaths = 65.29194139194139
```

```
std_cases = covid_data['cases'].std()
std_deaths = covid_data['deaths'].std()
print("The standard_deviation_cases =",std_cases)
print('The standard_deviation_deaths =',std_deaths)
```

```
The standard_deviation_cases = 6490.510073102111
The standard_deviation_deaths = 113.95663405806982
```

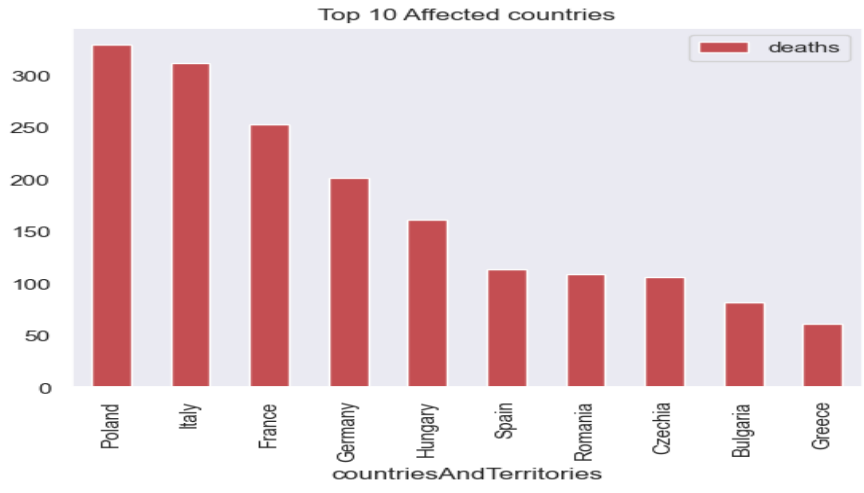
The mean (average) represents the central tendency of the data. In this context, it shows that, on average, there were approximately 3,661 reported COVID-19 cases and 65.29 reported deaths per day in the dataset.

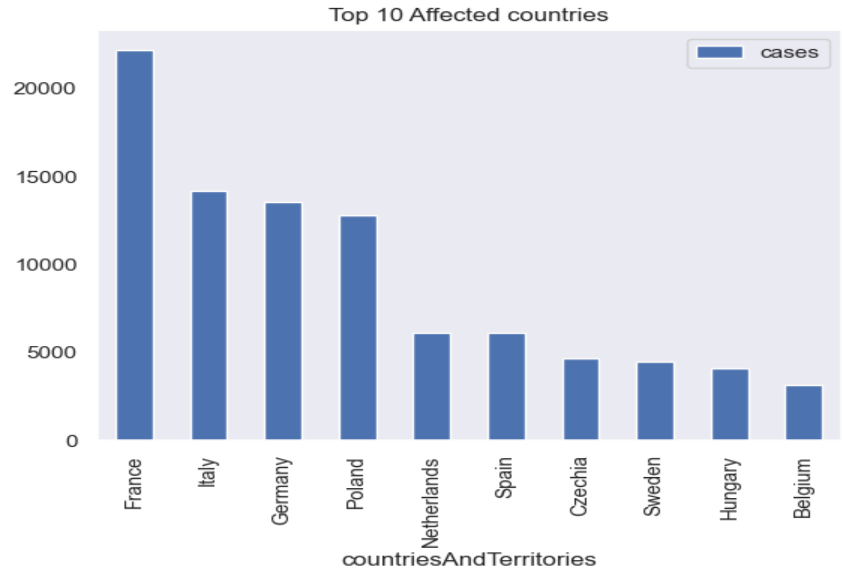
For cases: The standard deviation of 6,490.51 suggests that the number of daily COVID-19 cases varies considerably around the mean. Some days may have significantly higher case numbers, while others may have lower numbers.

For deaths: The standard deviation of 113.96 implies that the daily COVID-19 death counts also exhibit variation around the mean. There may be days with significantly more or fewer deaths.

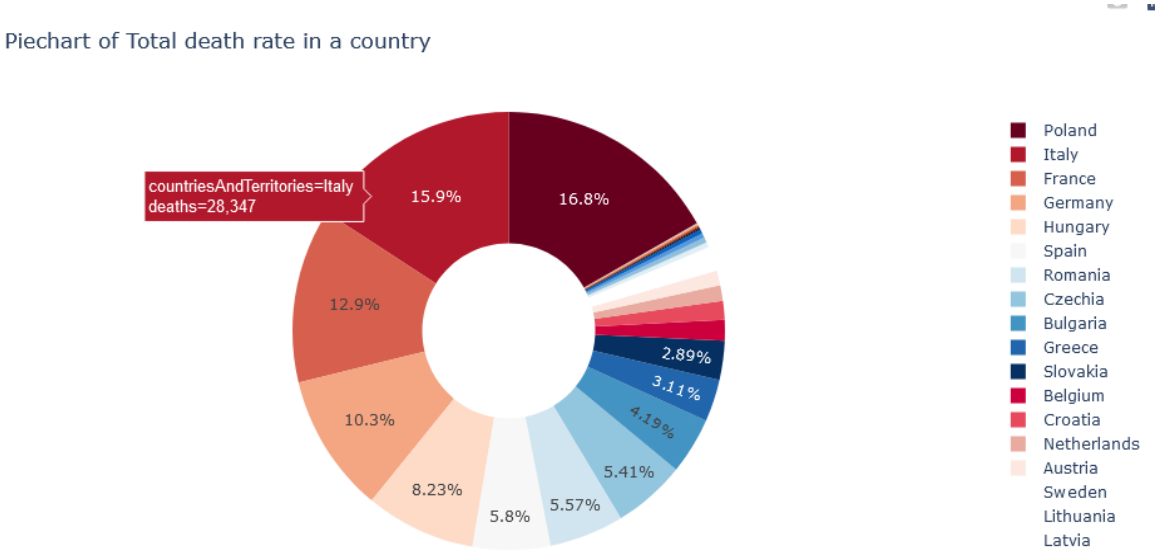
6.Visualization Using PYTHON Code:

```
# Importing all the important libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.colors as mcolors
import random
import math
import time
from sklearn.model_selection import RandomizedSearchCV, train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error, mean_absolute_error
import datetime
import operator
plt.style.use('seaborn')
%matplotlib inline
```

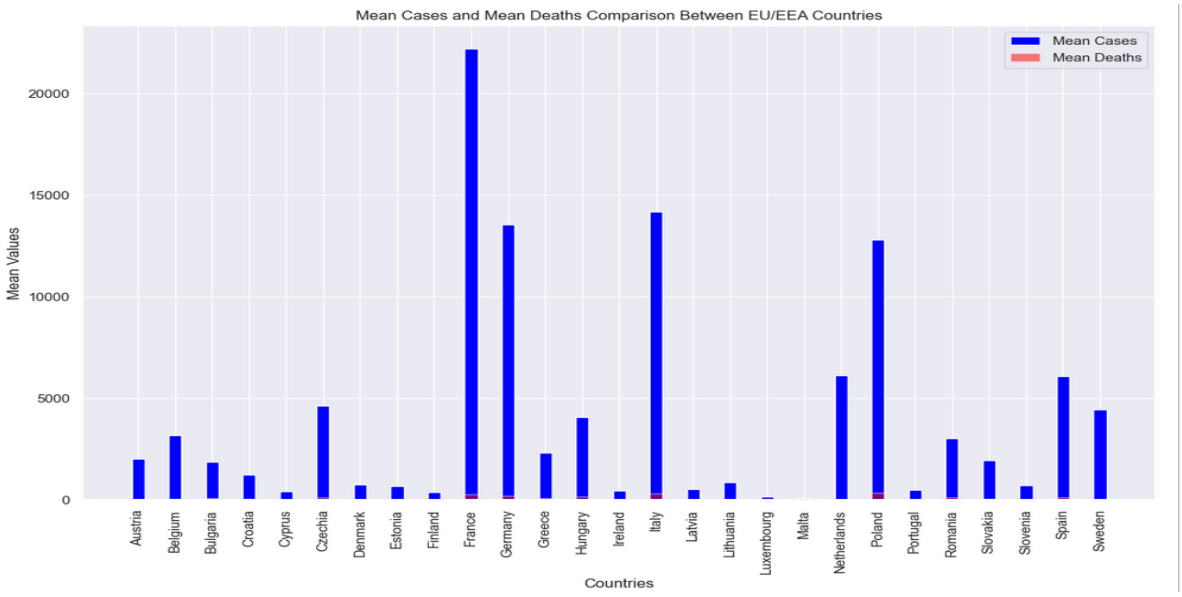




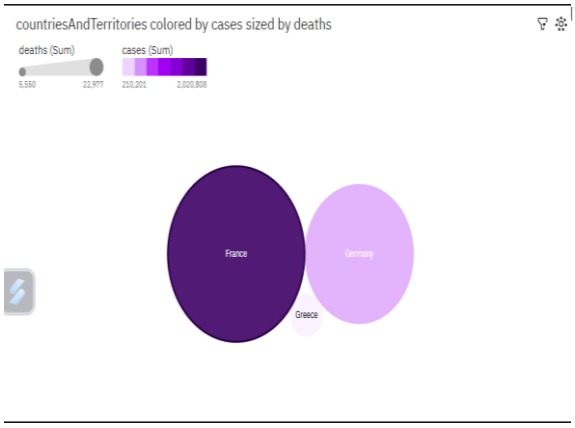

```
fig=plt.figure(figsize=(45,30))
fig=px.pie(confirmed_deaths,values='deaths', names='countriesAndTerritories'
           ,title='Piechart of Total death rate in a country', color_discrete_sequence=px.colors.sequential.RdBu,hole=.4)
fig.update_traces(textposition='inside')
fig.update_layout(uniformtext_minsize=12, uniformtext_mode='hide')
```

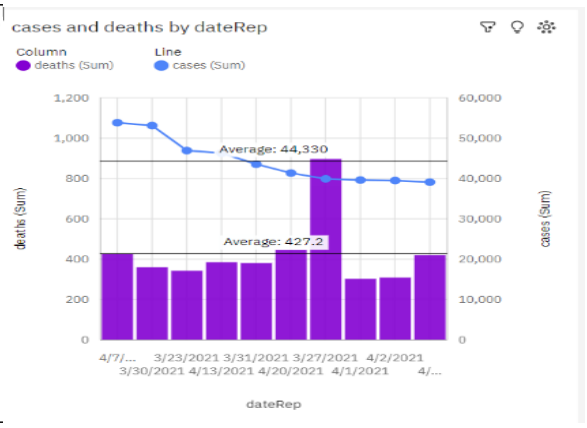


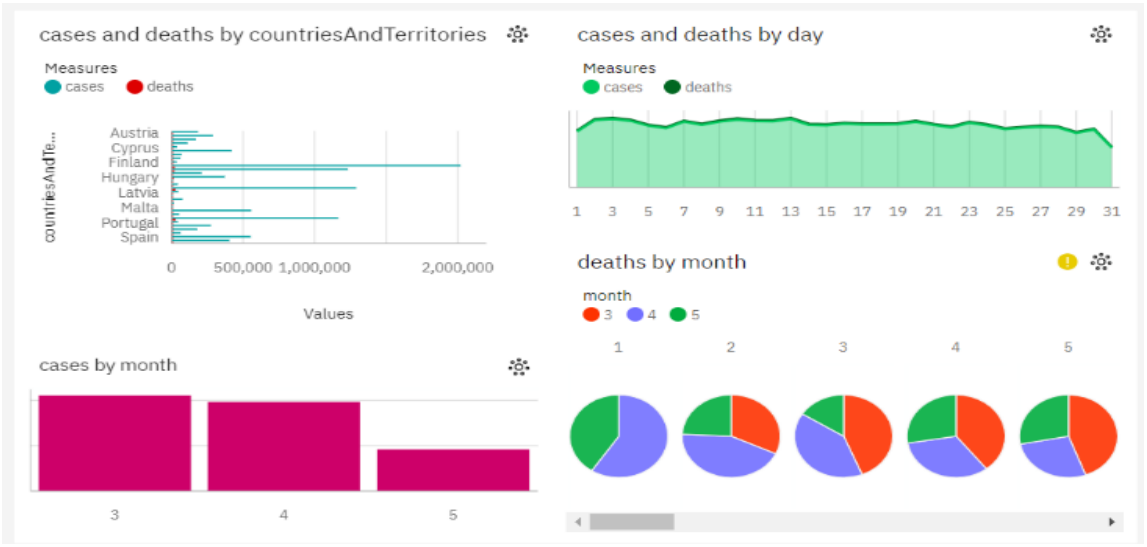
```
eu_countries = ['Austria', 'Belgium', 'Bulgaria', 'Croatia', 'Cyprus', 'Czechia', 'Denmark', 'Estonia', 'Finland', 'France', 'Ge

eu_data = covid_data[covid_data['countriesAndTerritories'].isin(eu_countries)]

# Calculate the mean values of cases and deaths for each country
summary_stats = eu_data.groupby('countriesAndTerritories').agg({'cases': 'mean', 'deaths': 'mean'}).reset_index()

# Create a bar chart to compare mean values between each pair of countries
fig, ax = plt.subplots(figsize=(12, 8))
countries = summary_stats['countriesAndTerritories']
mean_cases = summary_stats['cases']
mean_deaths = summary_stats['deaths']

bar_width = 0.35
index = range(len(countries))
plt.bar(index, mean_cases, bar_width, label='Mean Cases', color='blue')
plt.bar(index, mean_deaths, bar_width, label='Mean Deaths', color='red', alpha=0.5)

plt.xlabel('Countries')
plt.ylabel('Mean Values')
plt.title('Mean Cases and Mean Deaths Comparison Between EU/EEA Countries')
plt.xticks(index, countries, rotation=90)
plt.legend()

plt.tight_layout()
plt.show()
```

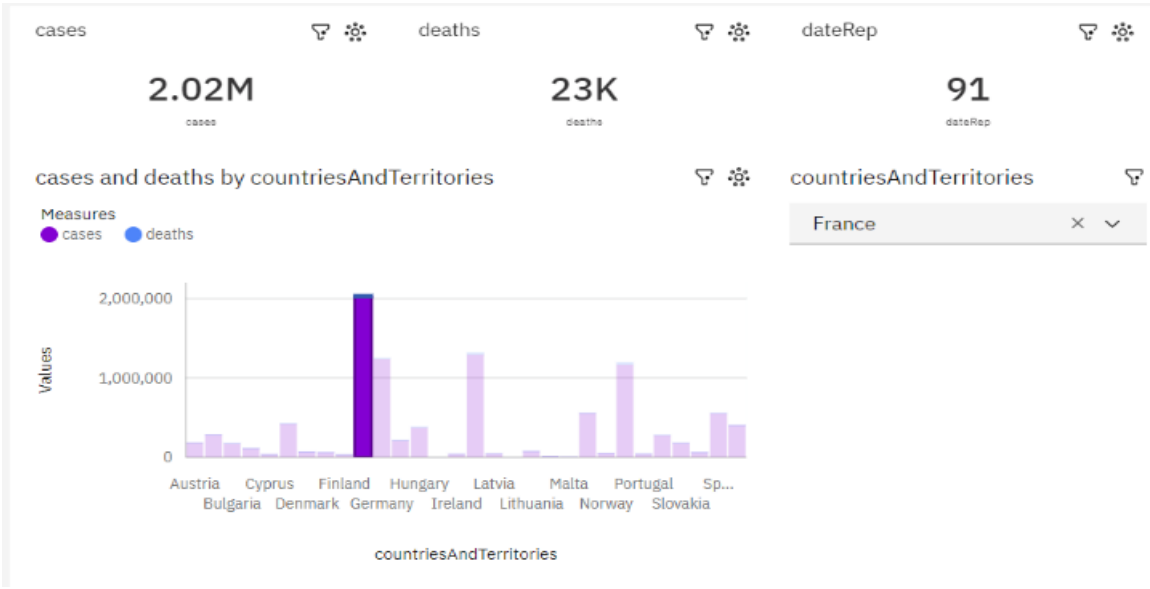


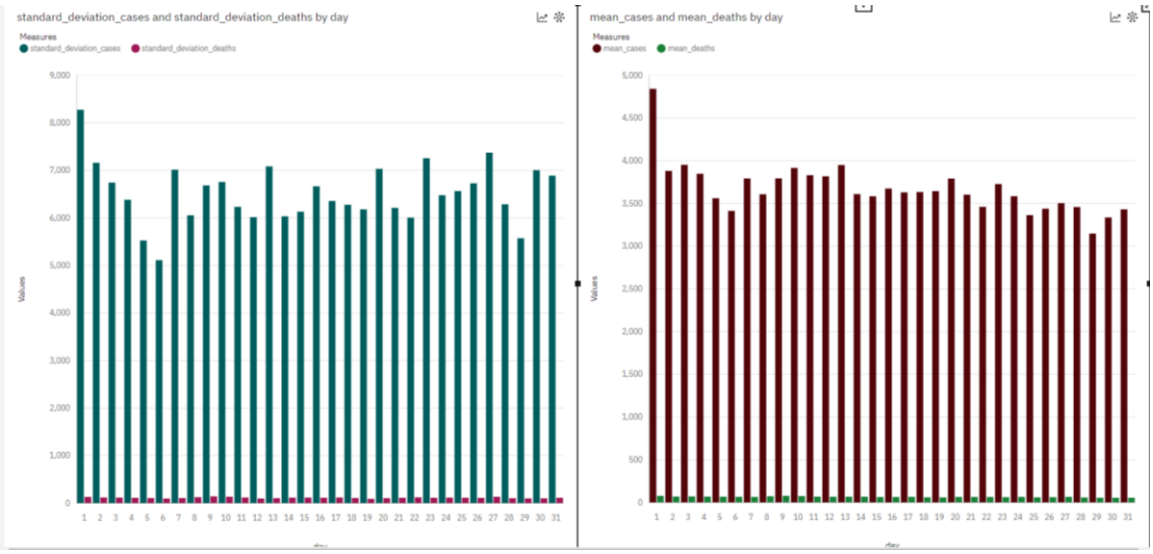
7.Visualization Using IBM Cognos:






cases and deaths by countriesAndTerritories
Measures
cases
deaths
countriesAndTe...
Austria
Cyprus
Finland
Hungary
Latvia
Malta
Portugal
Spain
0
500,000
1,000,000
2,000,000
Values
cases and deaths by day
Measures
cases
deaths
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
deaths by month
month
3
4
5
1
2
3
4
5
cases by month
3
4
5


cases
2.02M
cases
deaths
23K
deaths
dateRep
91
dateRep
cases and deaths by countriesAndTerritories
Measures
cases
deaths
2,000,000
1,000,000
0
Values
Austria
Cyprus
Finland
Hungary
Latvia
Malta
Portugal
Sp...
Bulgaria
Denmark
Germany
Ireland
Lithuania
Norway
Slovakia
countriesAndTerritories
countriesAndTerritories
France


standard_deviation_cases and standard_deviation_deaths by day
Measures
standard_deviation_cases
standard_deviation_deaths
Values
day
mean_cases and mean_deaths by day
Measures
mean_cases
mean_deaths
Values
day

The total number of results for deaths, across all countriesAndTerritories, is over 2500. cases ranges from 437, when countriesAndTerritories is Liechtenstein, to over 2.0 million, when countriesAndTerritories is France.

month 3 has the highest Total cases but is ranked 2 in Total deaths. Over all months, the sum of cases is nearly 10.0 million. cases ranges from over 1.8 million, when month is 5, to over 4.2 million, when month is 3.

For cases, the most significant values of month are 3 and 4, whose respective cases values add up to almost 8.2 million, or 81.7 % of the total. cases ranges from almost 206 thousand, when day is 31, to nearly 356 thousand, when day is 3.

8.Conclusion:

From the analysing the above visualizations, we can conclude that France has the highest Total cases but is ranked 3 in Total deaths. Poland has the highest Total deaths but is ranked 4 in Total cases. The hierarchy of highest death rate in countries/teritorries follows as Poland, Italy, France, Germany etc. The hierarchy of highest cases rate follows as France, Italy, Germany, Poland etc. In this project svm model have been used for regression and future prediction . The svm model has significantly lower MAE amd MSE values, indicating that it is better at predicting the target variable. The model R-squared value is close to 1, hence it has a strong fit.