ANALYSIS ON INDIAN ECONOMY

Project Report
on
Analysis on Indian Economy

SUPERVISOR: Mr. Mehtab Alam

SUBMITTED BY

Mihir 21001570052
Pranav Singh 21001570062
Yash Gupta 21001570099

2023-2024

Department of Computer Science
ACHARYA NARENDRA DEV COLLEGE
University of Delhi, Delhi-110019

ACKNOWLEDGEMENT

On the successful completion of our project Indian Economy, we extend our deepest appreciation and gratitude to our esteemed guide, Mr. Mehtab Alam, for his invaluable support, guidance, and mentorship throughout our data mining project. His expertise and insights have been instrumental in shaping the direction of our project, and his unwavering commitment to excellence has challenged us to perform at our best.

We would also like to express our heartfelt thanks to the faculty members of the Department of Computer Science for their encouragement and support. Their invaluable feedback, guidance, and suggestions have played a significant role in helping us improve the quality of our work.

We are deeply grateful for the opportunities provided to us, and we recognize that our success would not have been possible without the support and contributions of these individuals and institutions.

Mihir Pranav Yash

ACHARYA NARENDRA DEV COLLEGE
(University of Delhi)

CERTIFICATE

This is to certify that the data mining project titled "**Analysis on Indian Economy** " has been completed by Pranav Singh, Mihir and Yash Gupta who are pursuing Bachelor of Computer Science (Hons.) from Acharya Narendra Dev College, University of Delhi, during Semester-VI under the expert supervision of Mr. Mehtab Alam.

Supervisor
Mehtab Alam

Table of Contents

Chapter 1
PROBLEM STATEMENT

The Indian economy is a complex blend of various sectors, each exerting a distinct influence on its overall growth trajectory. Understanding the dynamic interplay between these sectors and their impact on the Gross Domestic Product (GDP) is essential for informed policymaking and strategic decision-making by businesses.

Our project aims to utilize data mining techniques to forecast India's annual GDP for the upcoming years. Additionally, we aim to uncover the intricate relationship between GDP fluctuations and the performance of key sectors within the Indian economy.

Key Objectives:

- GDP Prediction: Develop robust predictive models to forecast India's annual GDP for the forthcoming years. These models will utilize historical GDP data alongside relevant economic indicators and external factors that significantly influence economic growth.

- Sectoral Analysis: Conduct an in-depth analysis of various sectors contributing to the Indian economy, including agriculture, manufacturing, services, infrastructure, and finance. Identify the sectors that exhibit the strongest correlations with GDP fluctuations.

- Correlation Study: Explore the correlation between GDP fluctuations and the performance indicators of each sector. This analysis will entail examining factors such as sectoral output, investments, employment rates, inflation, government policies, and global economic trends.

Identifying Key Drivers: Determine the primary drivers within each sector that exert the most influence on GDP growth. By identifying these drivers, we aim to provide insights into the strategic areas where policy interventions or business strategies can be most effective in fostering economic growth.

By addressing these objectives, our project seeks to contribute to a deeper

understanding of the dynamics of the Indian economy and facilitate evidence-based decision-making for stakeholders across various sectors.

Chapter 2
DATA MINING TECHNIQUES

2.1. DM Techniques

2.1.1. Classification

Classification is a data mining technique that involves categorizing data into predefined classes or groups based on certain attributes or features. It is used to predict the categorical class labels of new data instances based on past observations. In classification, algorithms learn from labeled training data to assign classes to unseen data based on their similarities to known data instances.

2.1.2. Association

Association is a data mining technique used to discover relationships or associations between variables in large datasets. It identifies patterns where one event is associated with another event based on their co-occurrence within the dataset. Association rule mining is commonly used to uncover hidden patterns, such as "if-then" relationships, which can be valuable for market basket analysis, recommendation systems, and understanding customer behavior.

2.1.3. Clustering

Clustering is a data mining technique that groups similar data points or objects together based on their characteristics or features. It is an unsupervised learning approach where algorithms automatically partition data into clusters or groups without prior knowledge of class labels. Clustering aims to find natural groupings within the data, enabling insights into the underlying structure and patterns present in the dataset.

2.2. Classification

2.2.1. K-NN

K-NN is a simple and intuitive algorithm used for classification and regression tasks. It works by finding the K closest data points in the feature space and making predictions based on their labels (for classification) or their average (for regression).

2.2.2. Naive Bayes

Naive Bayes is a probabilistic algorithm based on Bayes' theorem, with the "naive" assumption of independence among features. Despite its simplifying assumption, it's effective in many real-world situations, especially in text classification and spam filtering.

2.2.3. Decision Tree

Decision trees are versatile supervised learning algorithms capable of performing both classification and regression tasks. They work by partitioning the feature space into regions and making predictions based on simple decision rules inferred from the data.

2.3. Regression

2.3.1. Linear Regression

Linear regression is a statistical method used for modeling the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to the observed data. It assumes a linear relationship between the independent and dependent variables, allowing us to predict the target variable based on the values of the independent variables.

Chapter 3
Dataset Description

3.1 Dataset

The dataset comprises a range of economic indicators sourced from the World Bank, offering insights into the economic landscapes of various countries. These indicators encompass demographic factors such as total population and urban population growth rates, alongside key economic metrics like GDP (measured in current US dollars) and inflation rates. Sectoral contributions to GDP, including agriculture, forestry, fishing, and industrial activities, are also delineated as percentages of total GDP. Moreover, the dataset includes trade-related statistics such as exports and imports of goods and services, as well as indicators of economic stability and development, such as gross capital formation and external debt stocks. Additionally, fiscal parameters like tax revenue and military expenditure as percentages of GDP provide insights into government spending patterns.

3.1.1. Number of Records

```
num_records = len(gdp)
print("Number of Records:", num_records)
```

```
Number of Records: 63
```

3.1.2 Number of Attributes

```
num_attributes = len(gdp.columns)
print("Number of Attributes:", num_attributes)
```

```
Number of Attributes: 16
```

3.1.3. Types of Attributes

```
attribute_types = gdp.dtypes
print("Types of Attributes:\n", attribute_types)
```

```
Types of Attributes:
 Series Name
Population, total                                          float64
Urban population growth (annual %)                         float64
GDP (current US$)                                          float64
Inflation, GDP deflator (annual %)                         float64
Agriculture, forestry, and fishing, value added (% of GDP) float64
Industry (including construction), value added (% of GDP)  float64
Exports of goods and services (% of GDP)                   float64
Imports of goods and services (% of GDP)                   float64
Gross capital formation (% of GDP)                         float64
Revenue, excluding grants (% of GDP)                       float64
Tax revenue (% of GDP)                                     float64
Military expenditure (% of GDP)                            float64
Merchandise trade (% of GDP)                               float64
External debt stocks, total (DOD, current US$)             float64
Net barter terms of trade index (2015 = 100)               float64
Foreign direct investment, net inflows (BoP, current US$)  float64
dtype: object
```

3.1.4. Missing Values or Nulls

```
missing_values = gdp.isnull().sum()
print("Missing Values or Nulls:\n", missing_values)
```

```
Missing Values or Nulls:
 Series Name
Population, total                                                0
Urban population growth (annual %)                               1
GDP (current US$)                                                0
Inflation, GDP deflator (annual %)                               1
Agriculture, forestry, and fishing, value added (% of GDP)       0
Industry (including construction), value added (% of GDP)        0
Exports of goods and services (% of GDP)                         0
Imports of goods and services (% of GDP)                         0
Gross capital formation (% of GDP)                               0
Revenue, excluding grants (% of GDP)                            18
Tax revenue (% of GDP)                                          18
Military expenditure (% of GDP)                                  0
Merchandise trade (% of GDP)                                     0
External debt stocks, total (DOD, current US$)                  11
Net barter terms of trade index (2015 = 100)                    21
Foreign direct investment, net inflows (BoP, current US$)       10
dtype: int64
```

3.1.5. Attributes Description

```
attributes_description = gdp.describe()
print("Attributes Description:\n", attributes_description)
```

```
Attributes Description:
 Series Name  Population, total  Urban population growth (annual %)  \
count                   63.000                              62.000
mean             907933404.952                               2.984
std              306757727.771                               0.496
min              445954592.000                               2.025
25%              630487840.000                               2.599
50%              888941760.000                               2.922
75%             1181032768.000                               3.238
max             1417173120.000                               3.881

Series Name  GDP (current US$)  Inflation, GDP deflator (annual %)  \
count                   63.000                              62.000
mean          741431449437.460                               7.186
std           923682620231.784                               3.766
min            37029883904.000                              -1.649
25%           101121531904.000                               3.905
50%           296042070016.000                               7.749
75%          1069577502720.000                               8.907
max          3385089851392.000                              17.830
```

Series Name	Agriculture, forestry, and fishing, value added (% of GDP)
count	63.000
mean	27.755
std	9.282
min	16.032
25%	17.473
50%	27.585
75%	35.887
max	42.752

Series Name	Industry (including construction), value added (% of GDP)
count	63.000
mean	25.886
std	2.960
min	20.089
25%	23.801
50%	26.500
75%	27.590
max	31.137

Series Name	Exports of goods and services (% of GDP)
count	63.000
mean	11.238
std	7.217
min	3.308
25%	5.225
50%	8.494
75%	18.748
max	25.431

Series Name	Imports of goods and services (% of GDP)
count	63.000
mean	13.153
std	8.339
min	3.709
25%	6.618
50%	9.245
75%	21.083
max	31.259

Series Name	Gross capital formation (% of GDP)
count	63.000
mean	26.659
std	6.825
min	17.640
25%	19.806
50%	25.827
75%	30.970
max	41.951

Series Name	Revenue, excluding grants (% of GDP)	Tax revenue (% of GDP)
count	45.000	45.000
mean	12.040	9.764
std	0.937	0.997
min	9.422	8.079
25%	11.476	9.026
50%	11.960	9.689
75%	12.760	10.408
max	14.440	12.108

Series Name	Military expenditure (% of GDP)	Merchandise trade (% of GDP)	\
count	63.000	63.000	
mean	3.015	18.513	
std	0.492	10.770	
min	2.004	6.536	
25%	2.640	10.306	
50%	2.924	14.133	
75%	3.292	27.771	
max	4.231	43.035	

Series Name	External debt stocks, total (DOD, current US$)	\
count	52.000	
mean	165077259766.154	
std	181004248203.895	
min	8425121280.000	
25%	29722833920.000	
50%	94012682240.000	
75%	234411884544.000	
max	612865867776.000	

Series Name	Net barter terms of trade index (2015 = 100)	\
count	42.000	
mean	90.663	
std	11.533	
min	62.812	
25%	84.449	
50%	90.759	
75%	95.943	
max	113.933	

Series Name	Foreign direct investment, net inflows (BoP, current US$)
count	53.000
mean	13239358672.047
std	18721779920.537
min	-36060000.000
25%	73537640.000
50%	2168591104.000
75%	27396884480.000
max	64362364928.000

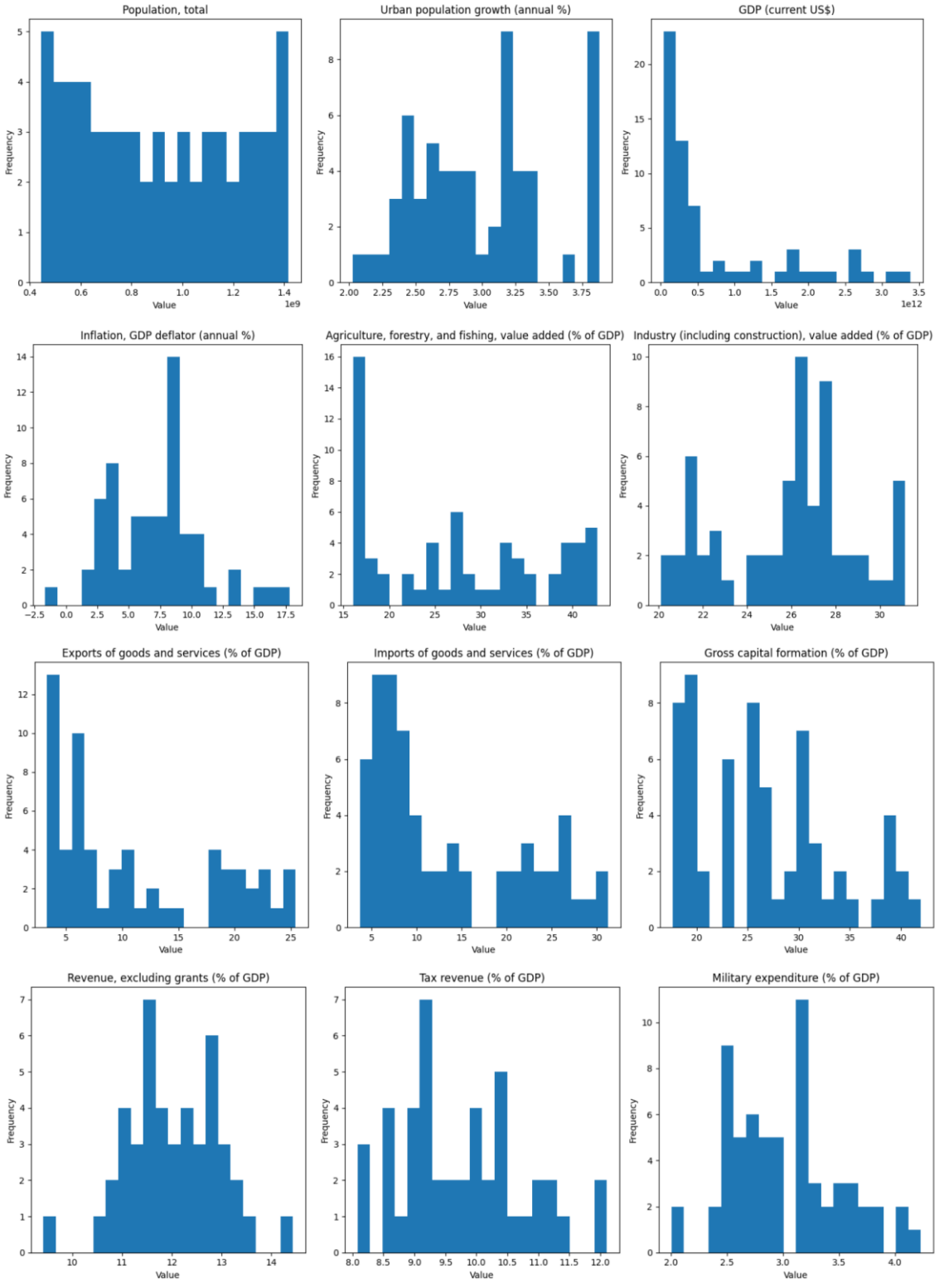
3.1.6. Distribution/Histograms

```
import matplotlib.pyplot as plt
num_rows = (len(gdp.columns) + 2) // 3
fig, axs = plt.subplots(num_rows, 3, figsize=(15, num_rows * 5))
axs = axs.flatten()

for i, column in enumerate(gdp.columns):
    ax = axs[i]
    ax.hist(gdp[column], bins=20)
    ax.set_title(column)
    ax.set_xlabel("Value")
    ax.set_ylabel("Frequency")

for j in range(len(gdp.columns), num_rows * 3):
    axs[j].axis('off')

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


Population, total
Urban population growth (annual %)
GDP (current US$)
Inflation, GDP deflator (annual %)
Agriculture, forestry, and fishing, value added (% of GDP)
Industry (including construction), value added (% of GDP)
Exports of goods and services (% of GDP)
Imports of goods and services (% of GDP)
Gross capital formation (% of GDP)
Revenue, excluding grants (% of GDP)
Tax revenue (% of GDP)
Military expenditure (% of GDP)
Frequency
Value
1e9
1e12

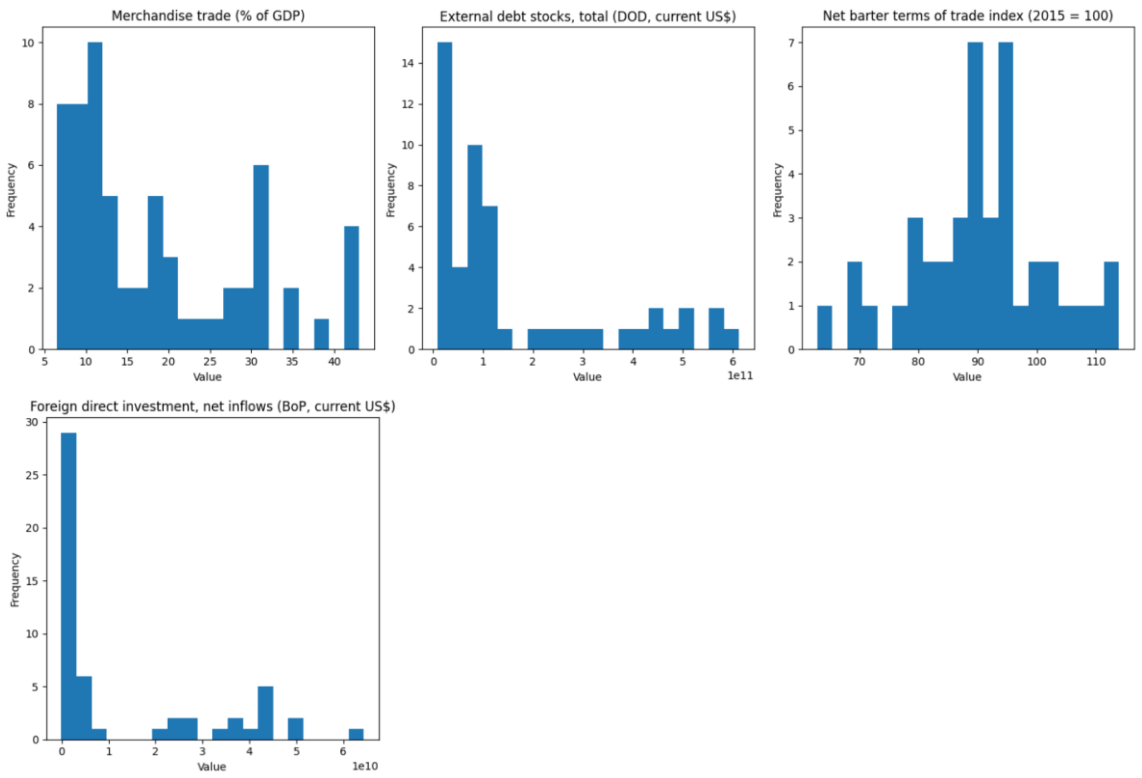


Figure 3.1: Histogram

3.1.7. Detecting Outliers

```
import seaborn as sns
num_rows = (len(gdp.columns) + 3) // 4
fig, axs = plt.subplots(num_rows, 4, figsize=(15, num_rows * 4), gridspec_kw={'hspace': 0.8})
axs = axs.flatten()

for i, column in enumerate(gdp.columns):
    ax = axs[i]
    words = column.split()
    title_lines = ['']
    for word in words:
        if len(title_lines[-1]) + len(word) + 1 > 15:
            title_lines.append('')
        title_lines[-1] += word + ' '
    sns.boxplot(x=gdp[column], ax=ax)
    ax.set_title('\n'.join(title_lines), fontsize=10)
    ax.set_xlabel("Value")

for j in range(len(gdp.columns), num_rows * 4):
    axs[j].axis('off')

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

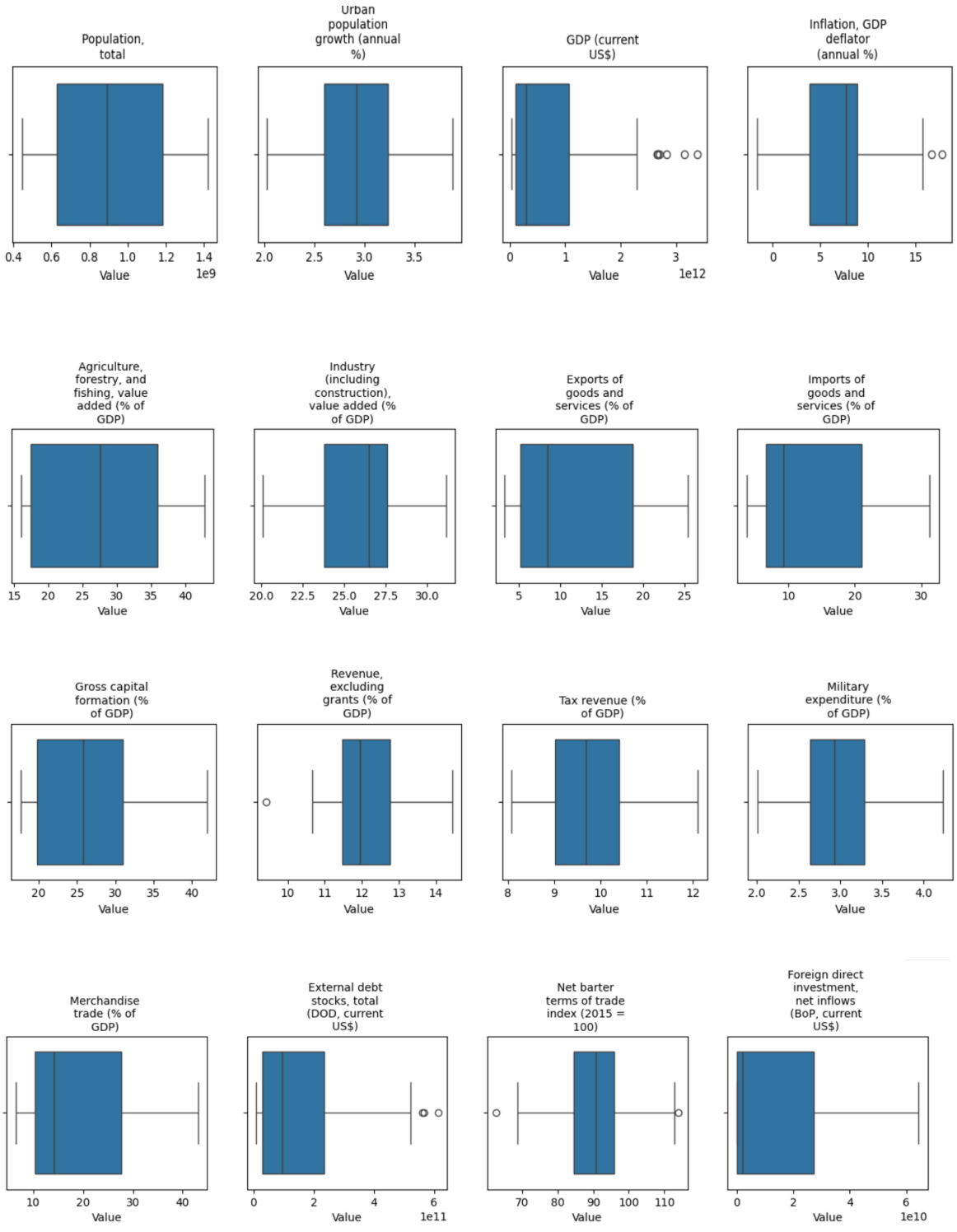


Figure 3.2: Outliers

Chapter 4
DATA PREPROCESSING

Data preprocessing plays a crucial role in the success of any data analysis project. It involves transforming raw data into a format that is suitable for analysis and model building. The importance of data preprocessing lies in improving data quality, reducing noise, handling missing values, and making the data suitable for the specific requirements of the analysis or model.

Data preprocessing involves several key components:

- Data Cleaning: Handling missing values, noisy data, and outliers by imputing or removing missing values, smoothing or correcting noisy data, and detecting and treating outliers appropriately.
- Data Transformation: Converting data into a suitable format for analysis or modeling by scaling numerical features, transforming skewed distributions, and encoding categorical variables into numerical formats using techniques like one-hot encoding or label encoding.
- Normalization and Standardization: Scaling numerical features to a similar range using normalization (0 to 1) or standardization (mean of 0 and standard deviation of 1) to prevent features with large scales from dominating the model training process.
- Dimensionality Reduction: Reducing the number of features in the dataset while preserving relevant information using techniques like principal component analysis (PCA) or feature selection methods to improve model performance, reduce computational complexity, and mitigate the curse of dimensionality.
- Data Splitting: Dividing the dataset into training, validation, and test sets before training machine learning models to ensure proper model training, hyperparameter tuning, and evaluation on unseen data.

4.1 HANDLING NULL VALUES

- Handling null or NaN (Not a Number) values is crucial in data preprocessing for several reasons:

1. Preventing Biased Analysis: Null values can bias analysis results if not handled properly. For example, if a significant portion of data is missing for a particular variable, omitting those observations can lead to biased conclusions.

2. Maintaining Data Integrity: Null values can disrupt data integrity and accuracy, affecting the quality of analysis and modeling. Leaving null values untreated may result in errors or inconsistencies in the results.

3. Improving Model Performance: Many machine learning algorithms cannot handle missing values directly. Therefore, handling null values appropriately is necessary to ensure the proper functioning and performance of machine learning models.

- Null values can arise due to various reasons:

1. Data Entry Errors: Human errors during data entry or collection processes may result in missing values.

2. Missing Information: Some information may not be available or may not be applicable for certain observations, leading to null values.

3. Data Collection Process: Incomplete data collection processes or technical issues can result in missing values.

4. Data Transformation: Null values can be generated during data transformation or manipulation processes, such as merging datasets or performing calculations.

- There are different ways to handle null values in a dataset:

1. Deletion: Removing observations or variables with null values. This approach is suitable when null values are relatively small in number and do not significantly affect the analysis.

2. Imputation: Filling in null values with estimated or calculated values. Common imputation techniques include replacing null values with the mean, median, mode, or a constant value, or using more advanced methods such as interpolation or predictive modeling.

3. Model-Based Imputation: Using regression models to predict missing values based on other variables in the dataset. This approach can provide more accurate imputations but requires additional computational resources and may introduce bias if the model assumptions are violated.

Below is a demonstration of how missing NaN values were handled in the project:

```
# Filling the missing NaN values
reversed_df = rd1.iloc[::-1]  # Reverse the DataFrame
interpolated_df = reversed_df.interpolate(method='linear')  # Interpolate missing values linearly
df = interpolated_df.iloc[::-1]  # Reverse the DataFrame back to original order
df.interpolate(inplace=True)  # Interpolate any remaining missing values
df.drop_duplicates()  # Remove duplicate rows, if any
```

Here, the interpolate() function from pandas library is used to fill in missing values linearly. Interpolation is a technique used to estimate missing values in a dataset by predicting intermediate values based on existing data points. It works by fitting a function to the observed data points and then using this function to estimate the values at missing points.

In our project, interpolation was used because our dataset contains GDP and other economic indicators measured over multiple years. There may be missing data points for certain years due to various reasons such as data collection issues or changes in reporting methods. Interpolation allows us to fill in these missing values by estimating them based on the observed data points before and after the missing values. By interpolating the missing values, we ensure that the temporal continuity of the data is preserved, enabling more accurate analysis and modeling of the economic trends over time. This approach is particularly useful when analyzing time-series data, where the chronological order of observations is essential for understanding the underlying patterns and trends in the data.

4.2 FEATURE SCALING

Feature scaling is a preprocessing technique used to standardize or normalize the range of independent variables or features in a dataset. It ensures that all features have a similar scale, which can improve the performance and convergence of machine learning algorithms, particularly those sensitive to the scale of input variables.

4.2.1 Normalization

Normalization scales the values of features to a range between 0 and 1. It is achieved by subtracting the minimum value of the feature and then dividing by the range (maximum value minus minimum value). This ensures that all feature

values fall within the same range, making them directly comparable. Normalization is particularly useful when the distribution of the feature values is not Gaussian or when the data has outliers.

4.2.2 Standardization

Standardization transforms feature values to have a mean of 0 and a standard deviation of 1. It is achieved by subtracting the mean of the feature and then dividing by the standard deviation. Standardization makes the distribution of feature values centered around 0, with a standard deviation of 1. This approach is suitable when the features are normally distributed and when the scale of the features is important for the model.

4.2.3 Feature Scaling in your project

In our project, we used the StandardScaler from the sklearn.preprocessing module to perform feature scaling. Below is a demonstration of how feature scaling was implemented in the project:

```
from sklearn.preprocessing import StandardScaler
columns_to_exclude = ['Growth Category', 'GDP Category', "GDP
(current US$)"]  # Add other columns as needed
columns_to_standardize = [col for col in gdp.columns if col not
in columns_to_exclude]

scaler = StandardScaler()
gdp_standardized = gdp.copy()
gdp_standardized[columns_to_standardize] =
scaler.fit_transform(gdp_standardized[columns_to_standardize])
gdp_standardized
```

4.3 FEATURE SELECTION AND CONVERSION

Feature selection is the process of selecting a subset of relevant features (variables, attributes) from a larger set of available features in a dataset. The goal of feature selection is to improve model performance, reduce overfitting, decrease computational complexity, and enhance interpretability by focusing on the most informative features while discarding irrelevant or redundant ones.

In our problem statement of predicting the annual GDP of the Indian economy and analyzing its correlation with other sectors, important features may include:

- GDP-related Indicators: Features directly related to GDP, such as Gross Domestic Product (GDP) itself, GDP growth rate, GDP per capita, etc. These indicators provide direct insights into the economic performance and growth of the Indian economy.

- Sector-specific Indicators: Features representing various sectors contributing to the Indian economy, such as agriculture, manufacturing, services, infrastructure, and finance. These indicators help understand the contribution of different sectors to GDP growth and their correlation with overall economic performance.

- Economic Indicators: Features capturing broader economic trends and conditions, such as inflation rate, unemployment rate, consumer spending, government expenditure, trade balance, foreign direct investment (FDI), etc. These indicators provide context and external factors influencing GDP growth.

4.3.1 CONVERSION FROM NUMERICAL TO CATEGORICAL

For the project, we converted the 'GDP (current US$)' column, representing the Gross Domestic Product (GDP) of the Indian economy, into categorical labels such as 'Very Low GDP', 'Low GDP', 'Average GDP', 'High GDP', and 'Very High GDP'. We also considered converting the 'GDP Growth' column into categorical labels representing different growth categories.

The reason for this conversion is to facilitate the application of classification models. Categorizing GDP and GDP growth into discrete classes allows us to analyze the impact of different levels of economic performance on various sectors or outcomes. It simplifies the analysis by transforming continuous variables into interpretable categories, making it easier to understand the relationship between economic indicators and other variables of interest.

4.4 DATA SAMPLING AND SUBSETTING

Splitting the data into training and testing sets is a fundamental step in a model development. Firstly, it enables us to evaluate the model's performance by training it on one subset of the data and testing it on another. This evaluation helps us understand how well the model generalizes to unseen data, indicating if it has learned meaningful patterns or if it's overfitting. Additionally, splitting the data facilitates hyperparameter tuning, where optimal parameters for the model are selected based on performance on a validation set. This ensures that the model's parameters are fine-tuned for optimal performance. Furthermore, by assessing the model's performance on a separate testing set, we can estimate its ability to generalize to new, unseen data, which is crucial for

deploying the model in real-world scenarios.

4.4.1 WAYS OF SAMPLING DATA

1. Simple Random Sampling: Every individual or data point has an equal chance of being selected, and each selection is independent of the others. This method is straightforward but may not be suitable for large datasets.

2. Stratified Sampling: The population is divided into homogeneous subgroups or strata, and samples are randomly selected from each stratum. This ensures representation from each subgroup and can improve the accuracy of estimates compared to simple random sampling
.
3. Systematic Sampling: Samples are selected at regular intervals from an ordered list of the population. It is less prone to bias than simple random sampling and is easier to implement.

4. Cluster Sampling: The population is divided into clusters, and a random sample of clusters is selected. Then, all individuals within the selected clusters are included in the sample. This method is useful when it is difficult or impractical to obtain a complete list of the population.

5. Convenience Sampling: Samples are selected based on convenience or accessibility. While easy to implement, this method may introduce bias as it does not ensure representativeness of the population.

6. Probability Proportional to Size (PPS) Sampling: Samples are selected with probabilities proportional to their sizes or weights in the population. This ensures that larger segments of the population have a higher chance of being included in the sample.

4.4.2 TRAIN-TEST SPLIT

```
import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder

random_years =
gdp.index.get_level_values(1).unique().to_series().sample(n=10,
random_state=42)
```

```
train_data =
gdp.loc[~gdp.index.get_level_values(1).isin(random_years)]
prediction_data =
gdp.loc[gdp.index.get_level_values(1).isin(random_years)]

columns_to_use = ['Agriculture, forestry, and fishing, value
added (% of GDP)',
                  'Industry (including construction), value
added (% of GDP)',
                  'Exports of goods and services (% of GDP)',
                  'Imports of goods and services (% of GDP)',
                  'Gross capital formation (% of GDP)',
                  'Revenue, excluding grants (% of GDP)', 'Tax
revenue (% of GDP)',
                  'Military expenditure (% of GDP)',
'Merchandise trade (% of GDP)']

label_encoder = LabelEncoder()
train_data['GDP Category'] =
label_encoder.fit_transform(train_data['GDP Category'])

# Training data
X_train = train_data[columns_to_use]
y_train = train_data['GDP Category']
clf = DecisionTreeClassifier()
clf.fit(X_train, y_train)

# Prediction data
X_prediction = prediction_data[columns_to_use]
predicted_growth_category = clf.predict(X_prediction)

predicted_growth_category =
label_encoder.inverse_transform(predicted_growth_category)
prediction_data['Predicted Growth Category'] =
predicted_growth_category

prediction_data
```

- **Training Data:**

X_train

Decade	Year	(% of GDP)		GDP)	GDP)		GDP)			
1960-1969	1961	41.093	21.435	4.304	5.958	19.262	9.422	8.181	2.074	9.360
	1962	39.066	22.053	4.169	6.032	18.108	9.422	8.181	2.745	8.930
	1963	39.825	21.880	4.280	5.907	18.995	9.422	8.181	4.025	8.473
	1964	41.344	20.955	3.726	5.685	19.530	9.422	8.181	3.819	8.111
	1966	40.352	21.387	4.143	6.672	18.358	9.422	8.181	3.572	11.761
	1967	42.752	20.089	4.034	5.947	18.198	9.422	8.181	3.224	8.748
	1968	42.086	20.627	4.039	4.943	17.742	9.422	8.181	3.246	8.159
	1969	41.585	21.418	3.714	4.031	17.640	9.422	8.181	3.141	6.924
	1970	40.290	21.729	3.783	3.879	18.211	9.422	8.181	3.185	6.648

- **Testing Data / Predictions:**

Agriculture, forestry, and fishing, value added (% of GDP)	Industry (including construction), value added (% of GDP)	Exports of goods and services (% of GDP)	Imports of goods and services (% of GDP)	Gross capital formation (% of GDP)	Revenue, excluding grants (% of GDP)	Tax revenue (% of GDP)	Military expenditure (% of GDP)	Merchandise trade (% of GDP)	External debt stocks, total (DOD, current US$)	barter terms of trade index (2015 = 100)	Foreign direct investment, net inflows (BoP, current US$)	Growth Category	GDP Category	Predicted GDP Category
41.741	20.834	4.463	6.834	17.931	9.422	8.181	2.004	9.816	8425121280.000	68.735	45460000.000	Very Low Growth	Very Low GDP	Very Low GDP
39.385	21.655	3.308	5.212	20.057	9.422	8.181	3.871	7.598	8425121280.000	68.735	45460000.000	Average Growth	Very Low GDP	Very Low GDP
38.627	22.408	4.027	3.709	19.548	9.422	8.181	3.718	6.536	10029268992.000	68.735	17790000.000	Very High Growth	Very Low GDP	Very Low GDP
34.312	24.480	6.686	6.115	19.419	11.055	9.182	3.478	10.917	14586390528.000	68.735	-7706430.500	Low Growth	Low GDP	Low GDP
29.718	26.629	5.255	7.646	26.446	12.527	10.075	3.569	10.781	38934814720.000	77.706	106090000.000	Average Growth	Average GDP	Average GDP
25.199	27.912	10.385	11.544	22.759	12.238	9.232	2.473	18.083	93966073856.000	95.021	2426056960.000	Average Growth	Average GDP	Average GDP
19.592	27.474	14.948	15.645	30.840	11.853	9.108	2.678	21.642	118884622336.000	92.298	3681984768.000	Low Growth	High GDP	High GDP
16.558	26.500	18.792	21.951	30.982	12.727	11.387	2.531	28.255	511472599040.000	98.522	39966093312.000	Low Growth	Very High GDP	Very High GDP

Chapter 5
Building Models

Training and testing different models is essential for several reasons:

- Model Comparison: Training and testing different models allow for a comparative analysis of their performance metrics, such as accuracy, precision, recall, or F1-score, helping determine which model performs best for the specific task.

- Robustness Assessment: Testing multiple models helps assess the robustness of the chosen algorithm. By evaluating different models on the same dataset, we can determine if the performance of a particular algorithm is consistent across various settings and conditions.

- Model Selection: In some cases, there may be uncertainty about which algorithm is most appropriate for a given problem. Training and testing different models allow for empirical validation of model selection decisions. By comparing the performance of various models on a validation dataset, we can make informed decisions about which model to deploy in production.

- Ensemble Learning: Ensembling involves combining the predictions of multiple models to improve overall performance. Training and testing different models enable the creation of diverse base learners for ensemble methods like bagging, boosting, or stacking. Each base learner contributes unique insights, and ensembling helps mitigate individual model biases and errors, leading to improved predictive accuracy.

- Understanding Model Behavior: Testing different models provides insights into their behavior and internal mechanisms. By analyzing how different algorithms make predictions and interpret data, we can gain a deeper understanding of their strengths, limitations, and underlying assumptions.

5.1 Decision Tree

- Description: Decision trees are versatile supervised learning algorithms capable of performing both classification and regression tasks. They work by partitioning the feature space into regions and making predictions based on

simple decision rules inferred from the data.

- Application: In our model, we used decision trees for classification tasks. The algorithm was trained to predict GDP categories by learning from the given features and their corresponding labels.
- Features Used: The decision tree model utilized the following features for training and prediction:
    1. Population, total
    2. Urban population growth (annual %)
    3. Agriculture, forestry, and fishing, value added (% of GDP)
    4. Industry (including construction), value added (% of GDP)
    5. Exports of goods and services (% of GDP)
    6. Imports of goods and services (% of GDP)
    7. Gross capital formation (% of GDP)
    8. Revenue, excluding grants (% of GDP)
    9. Tax revenue (% of GDP)
    10. Military expenditure (% of GDP)
    11. Merchandise trade (% of GDP)
    12. External debt stocks, total (DOD, current US$)
    13. Net barter terms of trade index (2015 = 100)
    14. Foreign direct investment, net inflows (BoP, current US$)

5.2 KNN

- Description: K-NN is a simple and intuitive algorithm used for classification and regression tasks. It works by finding the K closest data points in the feature space and making predictions based on their labels (for classification) or their average (for regression).
- Application: In our model, we utilized the K-NN algorithm for classification tasks, specifically for categorizing GDP growth into different categories based on various features.
- Features Used: The decision tree model utilized the following features for training and prediction:
    1. Population, total
    2. Urban population growth (annual %)
    3. Agriculture, forestry, and fishing, value added (% of GDP)
    4. Industry (including construction), value added (% of GDP)
    5. Exports of goods and services (% of GDP)
    6. Imports of goods and services (% of GDP)

7. Gross capital formation (% of GDP)
8. Revenue, excluding grants (% of GDP)
9. Tax revenue (% of GDP)
10. Military expenditure (% of GDP)
11. Merchandise trade (% of GDP)
12. External debt stocks, total (DOD, current US$)
13. Net barter terms of trade index (2015 = 100)
14. Foreign direct investment, net inflows (BoP, current US$)

5.3 Naïve Bayes

- Description: Naive Bayes is a probabilistic algorithm based on Bayes' theorem, with the "naive" assumption of independence among features. Despite its simplifying assumption, it's effective in many real-world situations, especially in text classification and spam filtering.
- Application: We applied the Naive Bayes algorithm for classification tasks, particularly in predicting GDP categories based on the given features.
- Features Used: The decision tree model utilized the following features for training and prediction:
    1. Population, total
    2. Urban population growth (annual %)
    3. Agriculture, forestry, and fishing, value added (% of GDP)
    4. Industry (including construction), value added (% of GDP)
    5. Exports of goods and services (% of GDP)
    6. Imports of goods and services (% of GDP)
    7. Gross capital formation (% of GDP)
    8. Revenue, excluding grants (% of GDP)
    9. Tax revenue (% of GDP)
    10. Military expenditure (% of GDP)
    11. Merchandise trade (% of GDP)
    12. External debt stocks, total (DOD, current US$)
    13. Net barter terms of trade index (2015 = 100)
    14. Foreign direct investment, net inflows (BoP, current US$)

5.4 Linear Regression

- Description: Linear regression is a statistical method used for modeling the relationship between a dependent variable (target) and one or more independent variables (features) by fitting a linear equation to the observed

data. It assumes a linear relationship between the independent and dependent variables, allowing us to predict the target variable based on the values of the independent variables.

- Application: In our model, linear regression was utilized as the primary regression model for predicting annual GDP. By training the model on historical data containing various economic indicators (features) and their corresponding GDP values (target), we were able to create a linear equation that best fits the relationship between these variables. This linear equation allows us to make predictions about future GDP values based on the values of the economic indicators.
- Features Used: The decision tree model utilized the following features for training and prediction:
    1. Population, total
    2. Urban population growth (annual %)
    3. Agriculture, forestry, and fishing, value added (% of GDP)
    4. Industry (including construction), value added (% of GDP)
    5. Exports of goods and services (% of GDP)
    6. Imports of goods and services (% of GDP)
    7. Gross capital formation (% of GDP)
    8. Revenue, excluding grants (% of GDP)
    9. Tax revenue (% of GDP)
    10. Military expenditure (% of GDP)
    11. Merchandise trade (% of GDP)
    12. External debt stocks, total (DOD, current US$)
    13. Net barter terms of trade index (2015 = 100)
    14. Foreign direct investment, net inflows (BoP, current US$)

Chapter 6
MODEL EVALUATION AND RESULTS

6.1 METRICS

Models are compared based on various evaluation metrics to assess their performance and determine which one best suits the problem at hand. The choice of evaluation metrics depends on the nature of the problem, the characteristics of the dataset, and the goals of the analysis.

For classification problems, evaluation metrics such as accuracy, precision, recall, F1-score, ROC-AUC, and confusion matrix are commonly used. In regression problems, evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), and R-squared (R^2) are commonly used. These metrics measure the accuracy of the model's predictions and its ability to explain the variance in the target variable.

6.1.1 CONFUSION MATRIX

A confusion matrix provides a tabular summary of the model's predictions compared to the actual labels. It includes counts of true positives, true negatives, false positives, and false negatives, allowing for a detailed analysis of the model's performance.

6.1.2 ACCURACY

Accuracy measures the proportion of correctly classified instances out of the total instances. It is a straightforward metric and is suitable for balanced datasets where classes are evenly distributed.

6.1.3 PRECISION

Precision measures the proportion of true positive predictions out of all positive predictions. It focuses on the accuracy of positive predictions and is useful when the cost of false positives is high. Precision is calculated as TP / (TP + FP), where TP is true positives and FP is false positives.

6.1.4 RECALL

Recall measures the proportion of true positive predictions out of all actual positives. It focuses on capturing all positive instances and is useful when the cost of false negatives is high. Recall is calculated as TP / (TP + FN), where FN is false negatives.

6.1.5 F1-SCORE

F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is a single metric that considers both precision and recall, making it useful when there is an imbalance between classes. F1-score is calculated as 2 * (Precision * Recall) / (Precision + Recall).

6.2. EXPERIMENTAL RESULTS AND COMPARISON

6.2.1 Decsion Tree

```
Metrics for Decision Tree Model:
Accuracy: 0.9
Precision: 0.95
Recall: 0.9
F1 Score: 0.9
Confusion Matrix:
[[1 0 1 0 0]
 [0 1 0 0 0]
 [0 0 1 0 0]
 [0 0 0 3 0]
 [0 0 0 0 3]]
```

6.2.2 Naïve Bayes

```
Metrics for Naive Bayes:
Accuracy: 0.9230769230769231
Precision: 0.9615384615384616
Recall: 0.9230769230769231
F1 Score: 0.9304029304029305
Confusion Matrix:
[[2 0 0 0 0]
 [0 2 0 0 0]
 [0 0 1 0 0]
 [0 0 0 4 0]
 [0 0 1 0 3]]
```

6.2.3 Knn

```
Metrics for Knn Model:
Accuracy of KNN Classifier: 1.0
Precision of KNN Classifier: 1.0
Recall of KNN Classifier: 1.0
F1 Score of KNN Classifier: 1.0
Confusion Matrix of KNN Classifier:
[[2 0 0 0 0]
 [0 2 0 0 0]
 [0 0 1 0 0]
 [0 0 0 4 0]
 [0 0 0 0 4]]
```

Chapter 7
INFERENCES AND CONCLUSION

The experimental evaluation of various machine learning models, including K-NN, Decision Tree, and Naive Bayes, revealed nuanced performances across different iterations. Remarkably, the K-NN model emerged as the standout performer, consistently achieving a flawless accuracy score of 1.0 during each training session. This consistent success underscores the robustness and reliability of the K-NN algorithm in capturing patterns within the dataset. Conversely, the Decision Tree model exhibited slightly more variability, with accuracy fluctuating between 0.8 and 1.0 across different runs, yet settling predominantly around 0.9. Despite these minor fluctuations, the Decision Tree model demonstrated commendable stability and maintained a high level of accuracy throughout. Similarly, the Naive Bayes classifier consistently delivered strong performances, with accuracy consistently hovering around 0.93 across multiple iterations. This consistent performance indicates the model's ability to generalize well to unseen data. Furthermore, employing linear regression for predicting annual GDP showcased promising outcomes, with minimal differences observed between predicted and actual values and a low mean squared error of 0.01576. These results underscore the efficacy of linear regression in accurately forecasting economic indicators, highlighting its potential as a valuable tool in economic analysis and prediction.

In conclusion, the evaluation of various machine learning models, including K-NN, Decision Tree, Naive Bayes, and linear regression, highlights their diverse applications and performances in analyzing and predicting economic indicators, particularly GDP. The K-NN model demonstrated exceptional consistency and reliability, consistently achieving perfect accuracy across multiple iterations. The Decision Tree and Naive Bayes classifiers also exhibited strong performances, with accuracy scores consistently near 0.9 and 0.93, respectively. These classification models offer valuable insights into classifying GDP levels as low, average, or high based on input features. Furthermore, the linear regression model showcased promising capabilities in accurately predicting future GDP values, indicating its potential as a valuable tool for forecasting economic trends. Overall, the combined use of classification and regression models provides a comprehensive framework for understanding and predicting GDP dynamics, offering valuable insights for economic analysis and policy-making.

Reference

1. World Bank. ' GDP (current US$).' *World Development Indicators*, The World Bank Group, 2015, https://data.worldbank.org/country/IN
2. Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
3. Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.
4. Kesavaraj, G., & Sukumaran, S. (2013, July). A study on classification techniques in data mining. In *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)* (pp. 1-7). IEEE.