Forecasting Tourist Arrivals in Bhutan Using Machine Learning Techniques.

**Submitted by Group 4:**

**Dorji Tshering**

**Sonam Jamtsho**

**Tshering Yangki**

**Megnath Sharma**

**Sonam Deki**

**Contents**

[Introduction 3](#_Toc191399598)

[Data Collection 3](#_Toc191399599)

[Data Source 3](#_Toc191399600)

[Key Features in the Dataset 3](#_Toc191399601)

[Data Preprocessing & Analysis 4](#_Toc191399602)

[Handling Missing Data 4](#_Toc191399603)

[Feature Engineering 9](#_Toc191399604)

[4. Machine Learning Model 12](#_Toc191399605)

[Training Machine Learning Models 12](#_Toc191399606)

[1. Linear Regression Model 12](#_Toc191399607)

[2. Random Forest Model 12](#_Toc191399608)

[3. XGBoost Model 12](#_Toc191399609)

[Evaluation Model Performance 13](#_Toc191399610)

[Results 13](#_Toc191399611)

[5. Making Future Predictions 14](#_Toc191399612)

[Prediction Results 14](#_Toc191399613)

[6. Data Visualization 14](#_Toc191399614)

[7. Conclusion 16](#_Toc191399615)

[ANNEX 17](#_Toc191399616)

# Introduction

This project aims to utilize machine learning techniques to forecast tourist arrivals in Bhutan for the years 2025 and 2030. It involves gathering data, preprocessing it, selecting appropriate models, making predictions, and visualizing the results to provide valuable insights for policymakers, businesses, and stakeholder.

# Data Collection

## Data Source

The dataset provided by our resource person included global tourism statistics, from which we extracted and filtered the specific data related to Bhutan for use in our analysis.

## Key Features in the Dataset

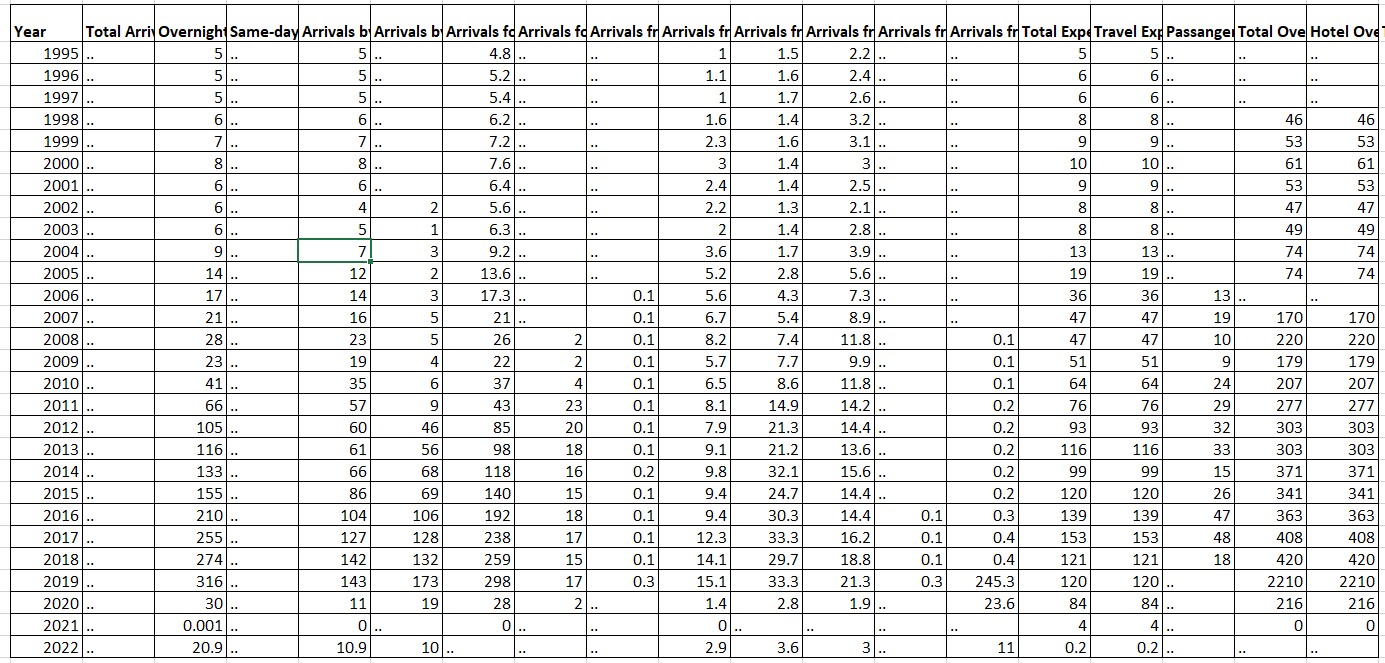
The dataset provides a comprehensive overview of Bhutan's tourism trends by categorizing visitors into overnight stays and same-day returns. It offers insights into tourists' transportation modes and visit purposes, such as leisure, business, or other reasons. The dataset also includes demographic details of tourists from different global regions. Additionally, it covers economic indicators like total expenditure, hotel stays, and transport spending, providing a thorough understanding of Bhutan's tourism landscape.

Figure 1: Dataset

# Data Preprocessing & Analysis

## Handling Missing Data

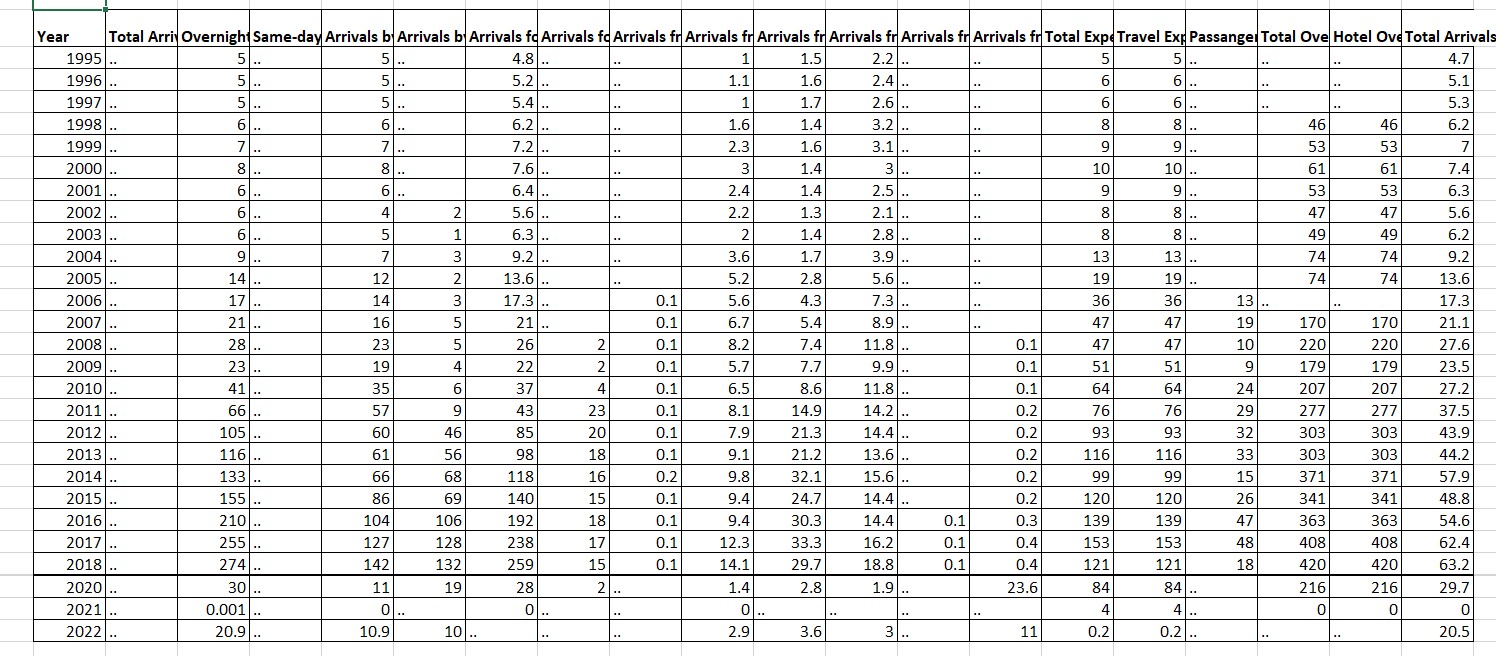
The dataset initially had a significant amount of missing data. We addressed this by first replacing missing values with NaN (Not a Number), and then filling all NaN values with 0 to ensure smooth numerical operations. Next, we converted string columns to numerical format; any values that couldn't be converted were replaced with NaN and subsequently filled with 0. We eliminated unnecessary columns, specifically "Total Arrivals (000s)" and "Same-day Visitors (000s)", as they lacked useful data. Finally, we created a new "Total Arrivals" column by summing the values from all regional arrival columns to represent region-based arrivals.

Figure 2: Dataset 2

We developed a correlation heatmap that visually illustrates the relationships between various numerical variables in the dataset, aiding in the identification of features with strong positive or negative correlations.

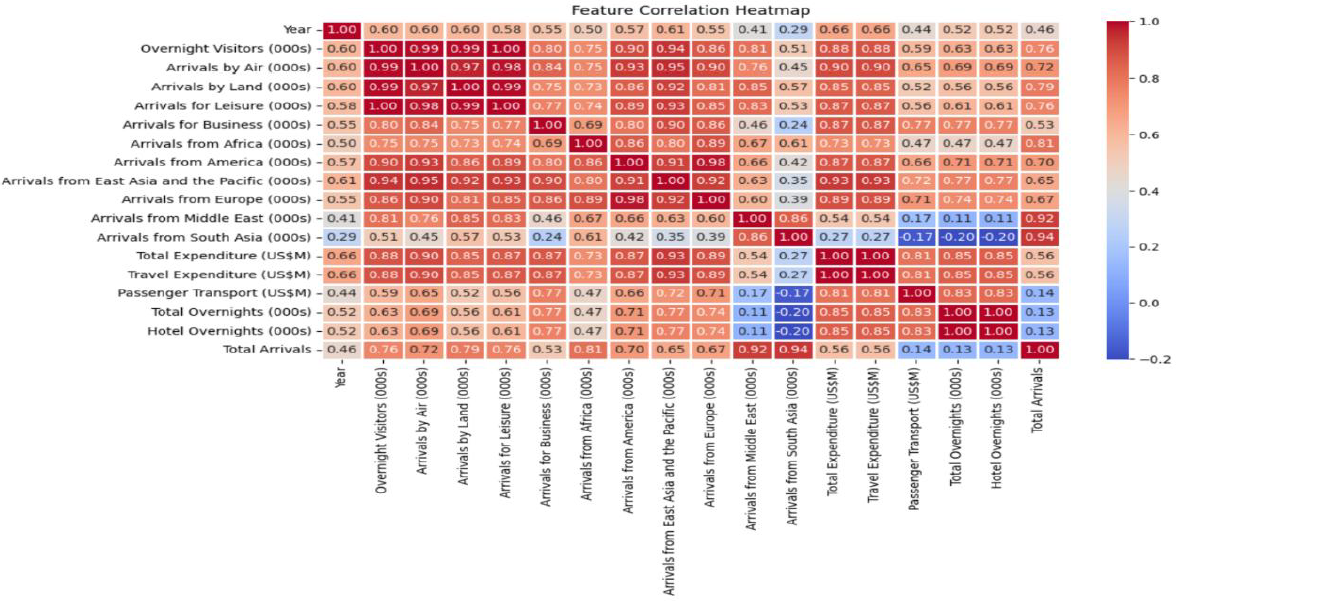
 Figure 3: Correlation heatmap

Figure 3 reveals a nearly perfect correlation between air arrivals and overnight visitors, with a coefficient of approximately 0.99. This suggests that the majority of tourists arriving by air choose to stay overnight. Additionally, leisure travelers significantly contribute to overnight stays, as evidenced by the strong correlation (~0.98) between leisure arrivals and overnight visitors.

There is also a notable correlation between tourist arrivals from Europe and America, likely due to similar travel patterns. Furthermore, total expenditure is largely driven by travel-related spending. However, tourist arrivals from South Asia have experienced fluctuations or declines over time, reflected in a negative correlation (~-0.29) with the year.

A line graph was created to visualize tourist arrivals from various regions, including Africa, America, East Asia and the Pacific, Europe, the Middle East, and South Asia. This graph highlights trends in tourist arrivals from these regions over time.

**Additional Insights:**

* **Regional Trends**: Bhutan's tourism is significantly influenced by regional visitors, particularly from India, which accounts for a large share of arrivals.
* **International Diversity**: Despite regional dominance, international arrivals are diverse, with key markets including the USA, UK, China, and others.
* **Economic Impact**: Tourism plays a crucial role in Bhutan's economy, with efforts to balance growth with sustainability

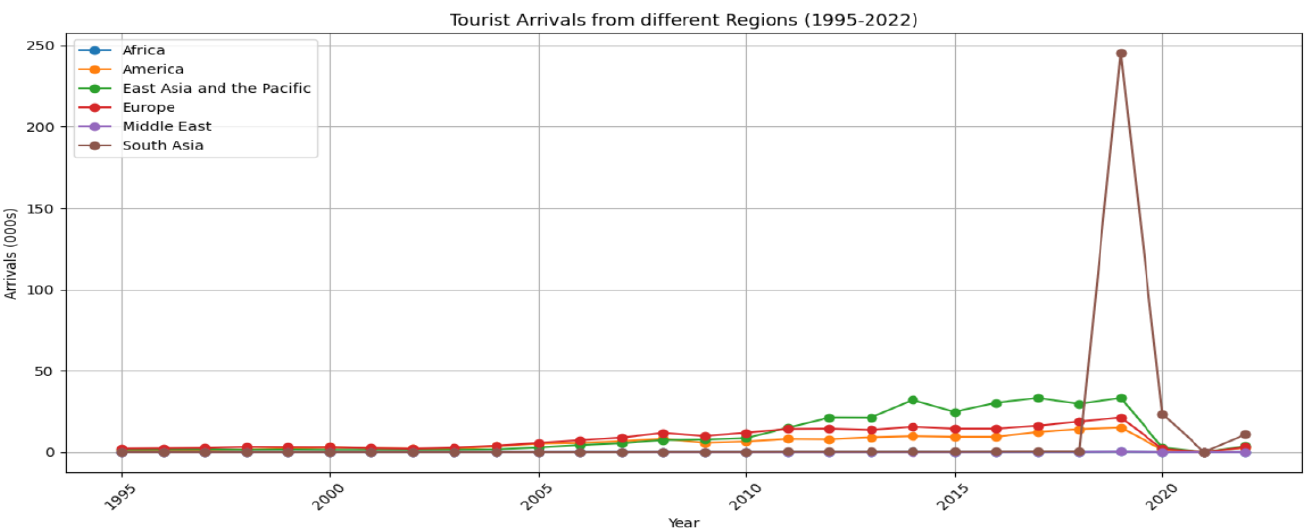
 Figure 4: Tourist Arrival from various regions

Figure 4 offers insights into the evolution of tourism trends over time, highlighting notable anomalies and shifts in travel behavior across different regions. The figure reveals a consistent increase in tourist arrivals from Europe, America, and East Asia & the Pacific over the years. However, there is a significant spike in arrivals from South Asia around 2020, followed by a sharp decline, which may suggest an unusual event or policy change affecting travel.

In contrast, tourist arrivals from Africa and the Middle East remain relatively low and stable throughout the period. Additionally, the figure shows a drastic drop in arrivals across all regions post-2020, likely due to the impact of COVID-19 travel restrictions.

A line graph was created to visualize the total tourist arrivals from 1995 to 2022, providing a comprehensive overview of these trends.

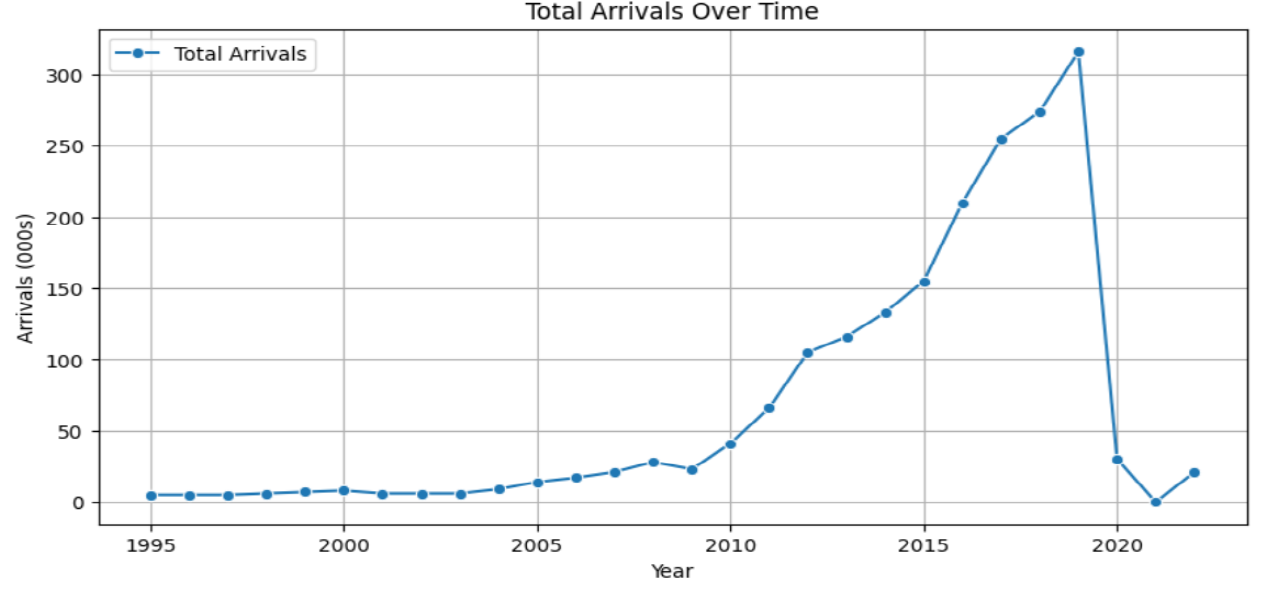


Figure 5: Total Tourist Arrival

Figure 5 illustrates that from 1995 to the late 2000s, tourist arrivals in Bhutan remained relatively stable with minimal fluctuations. However, around 2010, there was a significant surge in arrivals, followed by a consistent upward trend. This rapid growth likely reflects enhancements in tourism infrastructure, favorable policies, or increased global interest in visiting Bhutan. By 2019, tourist arrivals reached an all-time high of over 300,000. The COVID-19 pandemic led to a drastic decline in 2020, as travel restrictions severely impacted international tourism, causing arrivals to plummet. Following this low point, there was a gradual recovery from 2021 to 2022, indicating a slow reopening of borders and resumption of travel activities.

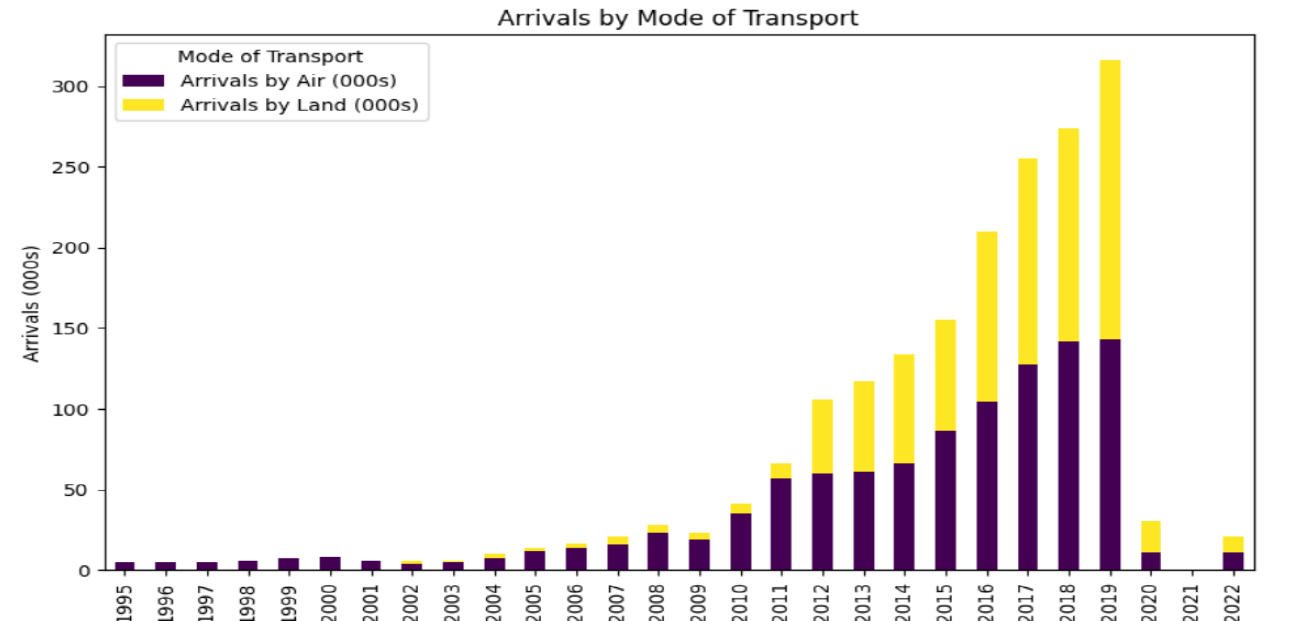


Figure 6: Mode of Transport

Tourist arrivals by transportation mode from 1995 to 2022 are depicted in Figure 6, highlighting arrivals by air (in purple) and by land (in yellow). From 1995 until the early 2000s, total arrivals were relatively modest, with air travel being the dominant entry method. A notable surge in total tourist arrivals occurred around 2010, accompanied by a significant increase in land arrivals, suggesting enhanced cross-border travel or a boost in regional tourism. From 2010 to 2019, both air and land arrivals showed consistent growth, with land arrivals making a considerable impact on total figures. This trend indicates improvements in transportation infrastructure, enhanced border access, or rising demand for regional tourism. The year 2019 saw a peak, with over 300,000 arrivals representing the highest level recorded.

In 2020, tourist arrivals saw a significant downturn, as both air and land travel faced a sharp decline attributed to COVID-19 travel restrictions. The number of arrivals dropped to a small percentage of earlier figures. After 2020, the graph indicates a modest recovery in 2021 and 2022, mainly in land-based arrivals, implying that cross-border travel rebounded more swiftly than air travel. Nonetheless, the overall number of arrivals continues to fall well short of pre-pandemic levels, reflecting a gradual and incomplete recovery in the tourism industry.

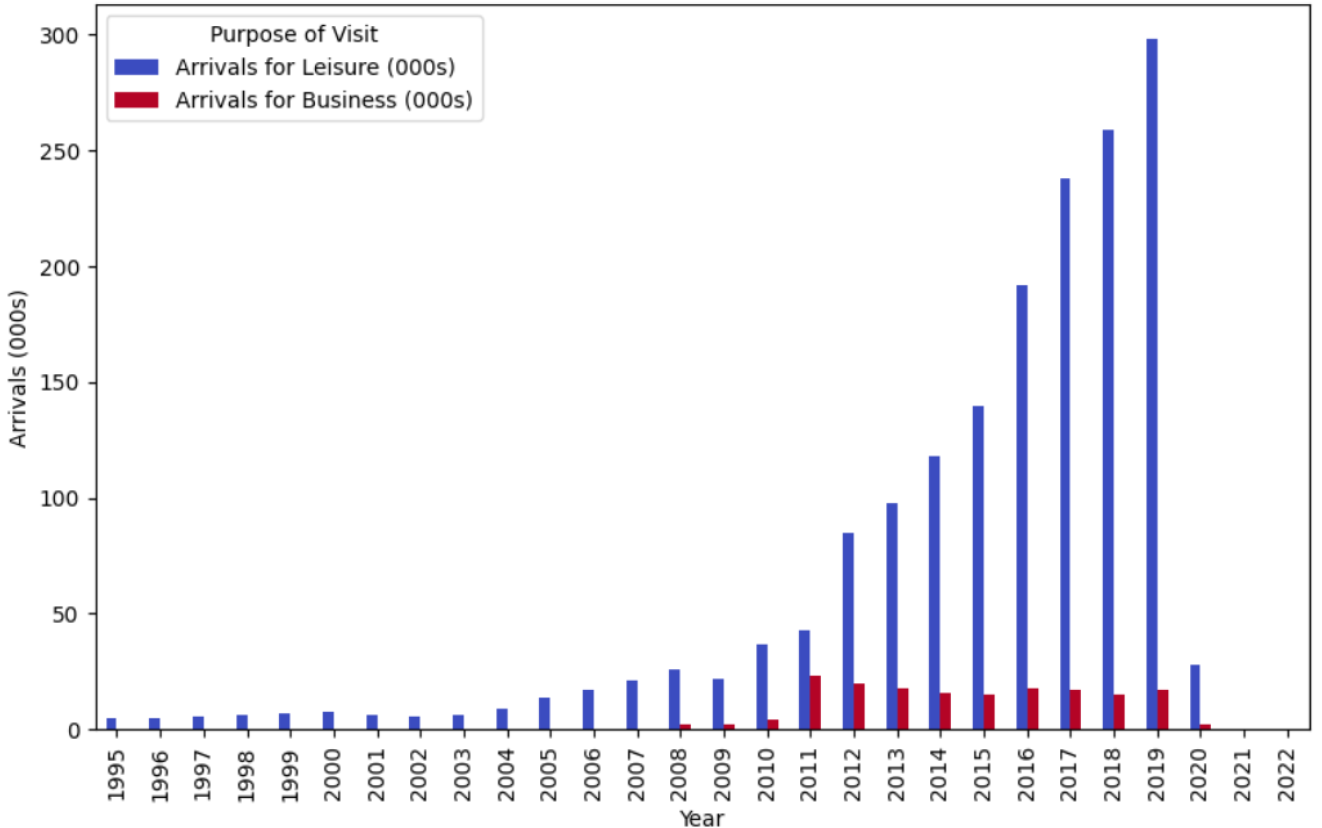


Figure 7: Arrival by Purpose

Figure 7 displays the number of arrivals (in thousands) for leisure and business purposes from 1995 to 2022, with leisure arrivals depicted in blue and business arrivals in red. From 1995 to approximately 2005, both leisure and business arrivals were relatively low and showed minimal fluctuation. However, beginning around 2006, leisure arrivals started to increase noticeably, with a sharp rise after 2010. This upward trend persisted steadily, culminating in a significant peak of nearly 300,000 arrivals in 2019. In contrast, business arrivals remained low throughout this period, experiencing only slight increases from 2010 to 2018. This pattern indicates that Bhutan has progressively embraced international tourism and development while seeing more restrained growth in business travel.

A marked decline in both leisure and business arrivals is apparent following 2019, with figures plummeting drastically in 2020, likely due to the worldwide effects of the COVID-19 pandemic. From 2020 onwards, arrivals seem to have stayed extremely low or virtually nonexistent. Overall, the graph underscores that leisure travel greatly surpassed business travel over the years, becoming the main contributor to the growth in total arrivals.

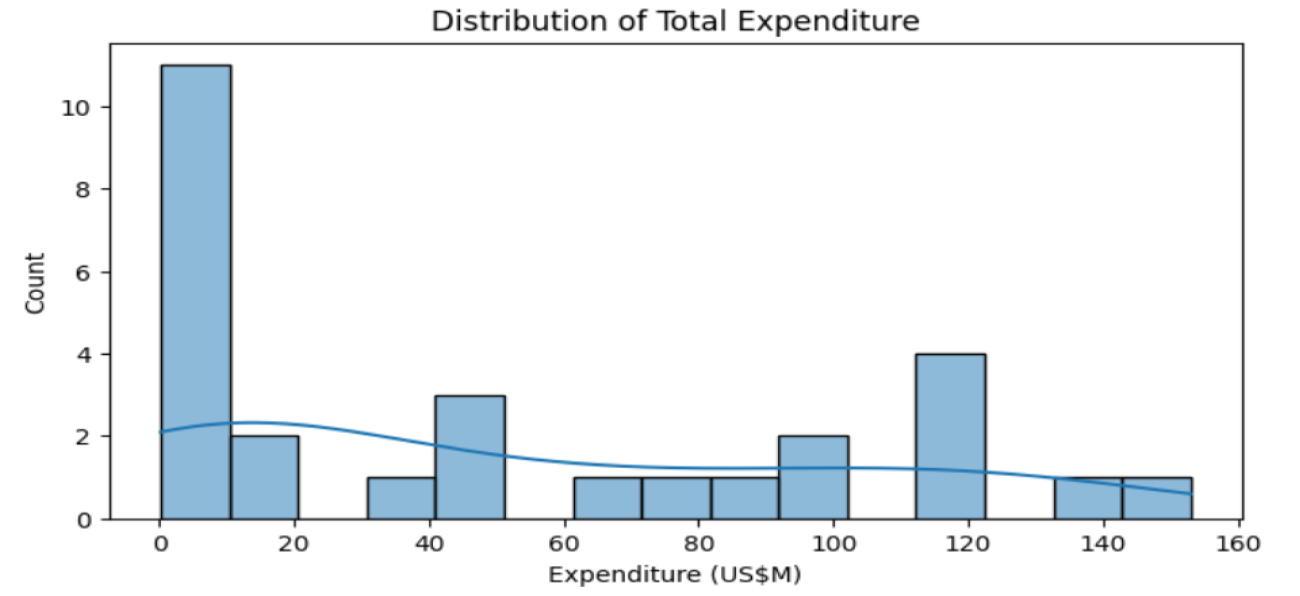


Figure 8: Distribution of expenditure

Figure 8 illustrates the distribution of total expenditures in millions of US dollars (USM). It shows that the majority of expenditures are concentrated in the lower range, with the highest frequency occurring in the first bin (0 20 USM). Its how’s that the majority of expenditures are concentrated in the lower range, with the highest frequency occurring in the first bin (0–20USM), which has over 10 occurrences, indicating that lower expenditures are the most prevalent. As expenditure amounts increase, the frequency tends to decline, resulting in a right-skewed distribution. However, there are notable spikes, especially in the 40–60 USMand100–120USM and 100–120USM ranges, suggesting that some higher expenditure amounts are also relatively common.The blue curve superimposed on the histogram represents a probability density function (a kernel density estimate), which aids in visualizing the overall distribution trend. This curve indicates that most expenditures are concentrated at the lower end, with a long tail extending toward higher expenditures. This distribution pattern suggests that while high expenditures do occur, they are significantly less frequent compared to smaller expenditures.

## Feature Engineering

Feature engineering plays a vital role in data analysis, enhancing the ability to identify trends and improve predictive modeling. In this context, several new features were developed to better understand tourism trends.

One key feature introduced is Year\_Squared, which captures non-linear growth patterns in tourist arrivals, allowing for more effective identification of periods of rapid growth or decline. To mitigate short-term fluctuations in the data, a 3-year moving average (MA\_Total\_Arrivals) was calculated, smoothing out seasonal variations and providing a clearer view of underlying trends.

Additionally, the Tourism Growth Rate was computed as the year-over-year percentage change in total arrivals. This metric reveals periods of significant growth, stagnation, or decline, offering insights into external factors such as economic shifts, policy changes, or global events like the COVID-19 pandemic.

To ensure a robust feature engineering process, a function called create\_features() was implemented. This function dynamically generates the necessary features while addressing missing or infinite values to maintain data consistency. It computes the moving average and percentage change safely, replacing infinite values with NaN and filling missing values with zeros to prevent errors in analysis.

Moreover, the function checks for the presence of essential columns, such as visitor types and expenditure data. If any required column is absent, it assigns the mean value from historical data or defaults to zero, ensuring that missing values do not hinder the analysis. Finally, any remaining infinite or NaN values are replaced with appropriate values to uphold data integrity.

Overall, this structured approach to feature engineering provides a solid framework for analyzing tourism trends, identifying patterns, and preparing data for predictive modeling.

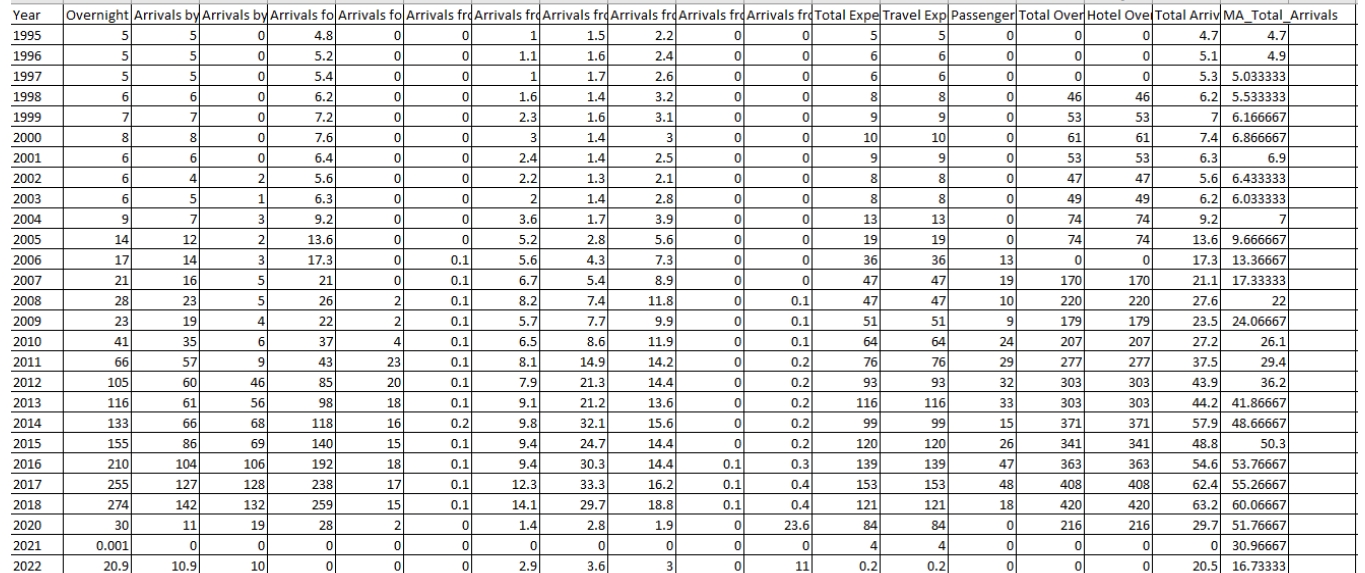


Figure 9: Final Dataset

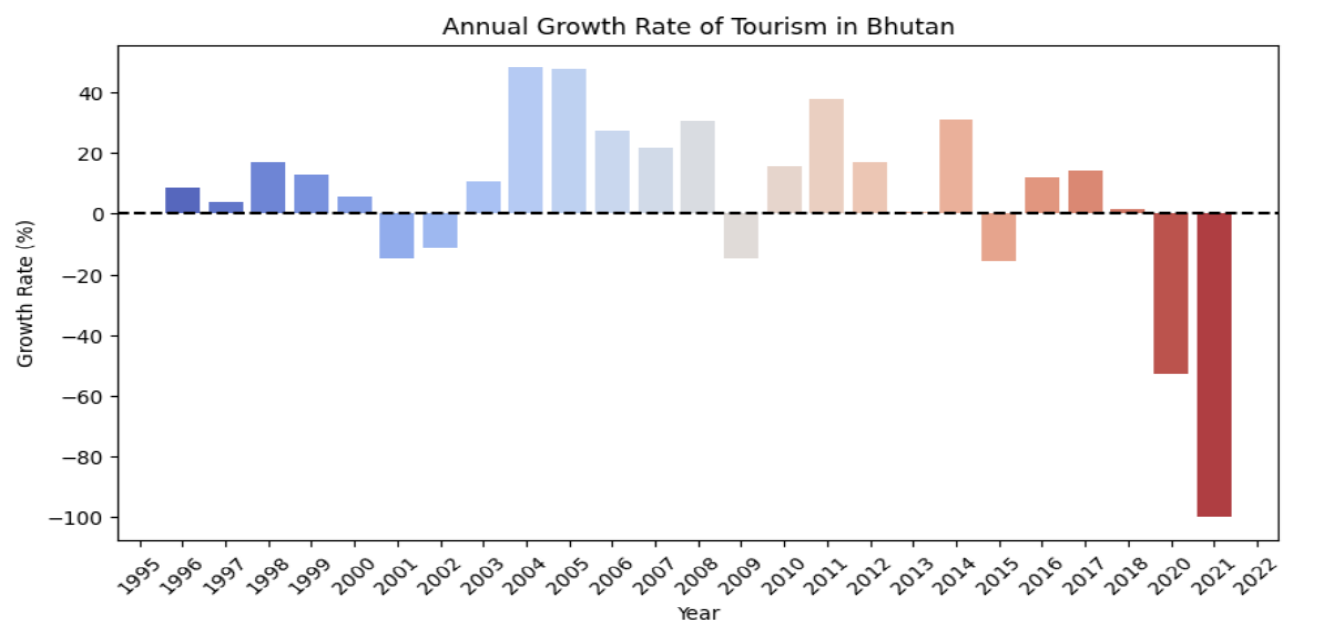


Figure 10: Annual Growth rate of Tourism

Figure 10 depicts the Annual Growth Rate of Tourism in Bhutan from 1995 to 2022. The x-axis represents the years, while the y-axis shows the growth rate percentage of tourism for each year. The dashed horizontal line at 0% marks the threshold between positive and negative growth.

**Key Insights:**

* Between 1995 and 2007, Bhutan experienced mostly positive growth in tourism, with some fluctuations. Notable peaks occurred around 2004–2005, with growth rates exceeding 40%.
* From 2010 to 2014, there was another phase of positive growth, with 2012 showing a particularly strong increase of around 40%.
* Declines occurred around 2000–2001 and again in 2009, likely due to external economic or political factors.
* A significant drop happened from 2019 to 2021. The decline in 2020 and the sharp plunge in 2021 (reaching nearly -100%) likely corresponds to the COVID-19 pandemic, which drastically affected global travel.

The data suggests that Bhutan's tourism industry has experienced periods of both growth and decline, with external factors such as economic conditions and global events playing a significant role in shaping the annual growth rate.

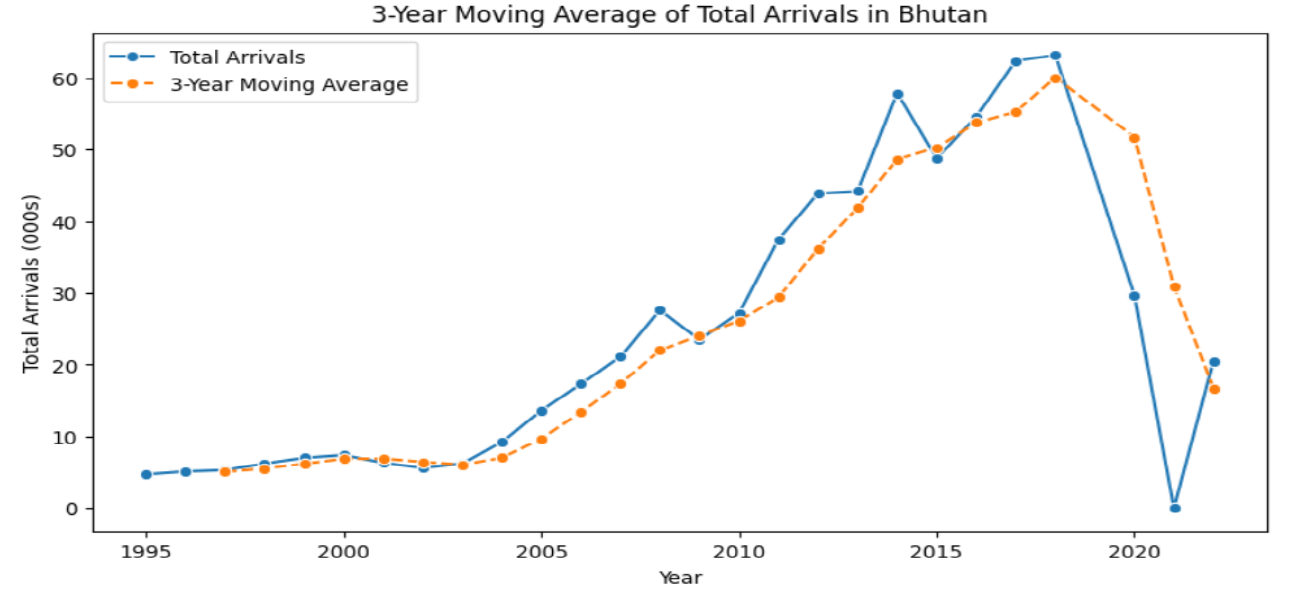


Figure 11: 3 years MA of Total Arrival

Figure 11 illustrates the trend of total tourist arrivals in Bhutan over time, smoothed by a 3-year moving average. From 1995 to the early 2000s, arrivals were low and stable. Starting around 2005, there was a steady increase, which accelerated after 2010 and peaked between 2018 and 2019 with nearly 315,600 arrivals, not 65,000 as mentioned. The moving average closely follows this trend, smoothing out short-term fluctuations. However, after 2019, arrivals sharply declined due to the COVID-19 pandemic, a downturn also reflected in the moving average, albeit with a slight lag.

# 4. Machine Learning Model

Building and training machine learning models for predicting tourism trends involves a crucial initial step: dividing the dataset into training and test sets. This ensures effective model training and unbiased performance evaluation on unseen data. Typically, 80% of the data is allocated for training, allowing the model to identify patterns, while the remaining 20% is used for testing to assess predictive accuracy. The train\_test\_split function from sklearn.model\_selection is used for this purpose.

The dataset is prepared by separating feature variables (X) from the target variable (y), which represents tourist arrivals, and removing non-predictive columns like "Year" and growth rates. The dataset is then split into training (X\_train, y\_train) and testing subsets (X\_test, y\_test), with 80% for training and 20% for testing. Setting random\_state=42 ensures that the split is reproducible, allowing consistent model comparisons.

This method ensures the model is trained on diverse data and evaluated separately, preventing overfitting and enhancing generalizability for real-world predictions.

# Training Machine Learning Models

To predict total tourist arrivals, several machine learning models are trained using various algorithms, each providing distinct strengths in identifying trends and patterns within the data.

## Linear Regression Model

The Linear Regression model is a straightforward and interpretable algorithm that assumes a linear relationship between input features and the target variable. It is initialized with LinearRegression() and trained on the dataset via lr\_model.fit(X\_train, y\_train). After training, it predicts outcomes on the test data using y\_pred\_lr = lr\_model.predict(X\_test). Although linear regression is effective for detecting general trends, it may struggle to capture complex non-linear relationships in the data.

## Random Forest Model

To enhance predictive performance, a Random Forest Regressor is also employed. This ensemble learning method builds multiple decision trees and aggregates their outputs to improve accuracy and robustness. The model is initialized with RandomForestRegressor(n\_estimators=100, random\_state=42), specifying 100 trees in the forest. It is trained on the dataset via rf\_model.fit(X\_train, y\_train) and generates predictions using y\_pred\_rf = rf\_model.predict(X\_test). Random Forest effectively handles non-linear relationships and mitigates overfitting by averaging the outputs of multiple trees.

## XGBoost Model

The XGBoost Regressor is also employed due to its efficiency and high predictive accuracy. XGBoost, an optimized form of gradient boosting, sequentially constructs trees to correct previous errors. It is initialized with XGBRegressor(n\_estimators=100, learning\_rate=0.1, random\_state=42), where n\_estimators=100 defines the number of boosting rounds and learning\_rate=0.1 controls the optimization step size. The model is trained using xgb\_model.fit(X\_train, y\_train) and makes predictions with y\_pred\_xgb = xgb\_model.predict(X\_test). XGBoost excels at handling large datasets with complex relationships.

# Evaluation Model Performance

To evaluate and compare the performance of various machine learning models, a function named evaluate\_model () is developed. This function computes four essential regression metrics:

* **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual values, with lower values indicating higher accuracy.
* **Mean Squared Error (MSE):** Calculates the average squared difference, penalizing larger discrepancies more than smaller ones.
* **Root Mean Squared Error (RMSE):** The square root of MSE, maintaining the same unit as the target variable and providing an interpretable error measure.
* **R² Score (Coefficient of Determination):** Indicates how well the model explains the target variable's variance, with scores closer to 1 signifying better performance.  
  These metrics were applied to assess the performance of the three models mentioned.

## Results

The predictions from each model are compared against the actual test values, and the results are used to identify which model offers the most accurate and reliable forecasts for total tourist arrivals.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **MSE** | **MAE** | **RMSE** | **R²** |
| Linear Regression | 0.00 | 0.00 | 0.00 | 1.00 |
| Random Forest | 0.48 | 0.67 | 0.82 | 1.00 |
| XGBoost | 2.74 | 11.33 | 3.37 | 0.96 |

Table 1: Evaluation Results

# 5. Making Future Predictions

To predict tourist arrivals for 2025 and 2030 using Linear Regression and Random Forest models, a new DataFrame named future\_years is created. The create\_features() function is applied to generate necessary features for these future years, ensuring consistency with the training data. The feature order is aligned with the training data, and any missing or infinite values are removed to maintain accuracy. Once the data is prepared, both models predict tourist arrivals for the specified years based on these features. The predicted values are then stored in the future\_years DataFrame and displayed, offering insights into future tourism trends based on historical patterns.

## Prediction Results

We predicted tourist arrivals for 2025 and 2030

|  |  |
| --- | --- |
| **Year** | **Predicted Arrivals(000s)** |
| 2025 | 24.296296 |
| 2030 | 24.296296 |

Table 2: Prediction Results

# 6. Data Visualization

We used Matplotlib and Seaborn to visualize the actual versus predicted tourist arrivals through line charts, comparing historical data with forecasted values.

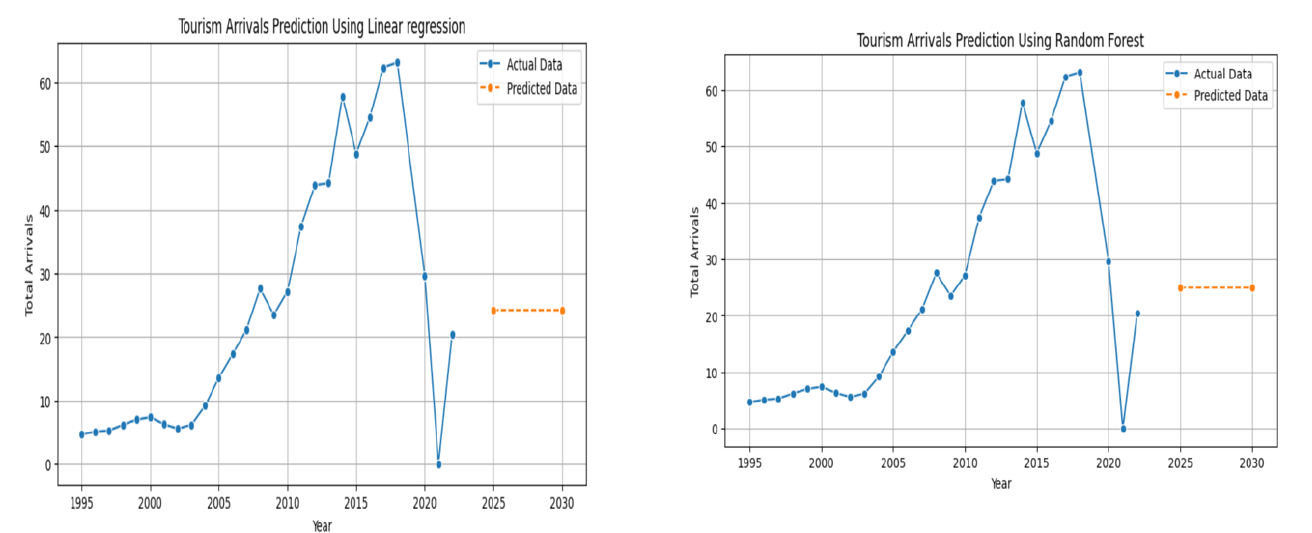


Figure 12: Line graph Tourist Arrival Prediction

Figure 12 presents future predictions of tourist arrivals in Bhutan using Linear Regression and Random Forest models. The graphs display historical data from 1995 to around 2022, followed by predicted values for 2025 and 2030. Historically, arrivals rose steadily until peaking in 2018-2019, then declined sharply due to the pandemic.

Predictions suggest a partial recovery by 2025 and 2030, though not reaching pre-pandemic levels. The consistent predictions from both models indicate a stabilized trend, suggesting a moderate and steady recovery in Bhutan's tourism sector. This insight can aid policymakers and stakeholders in preparing for future tourism developments

Scatter plots analyzing feature relationships.

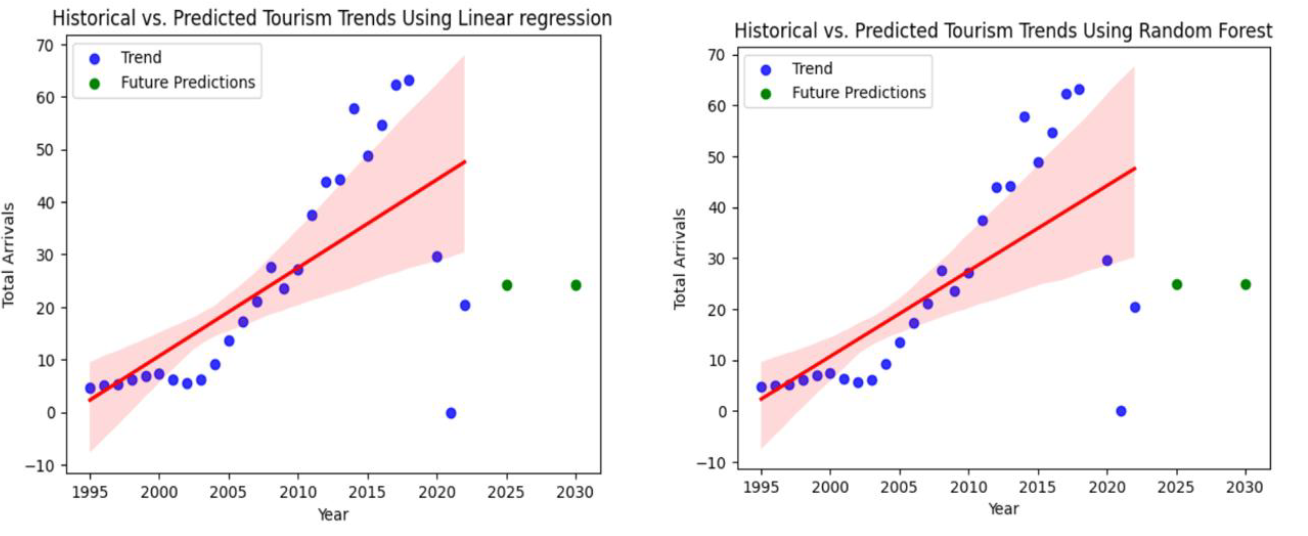


Figure 13. Scatter plots Tourist Arrival prediction

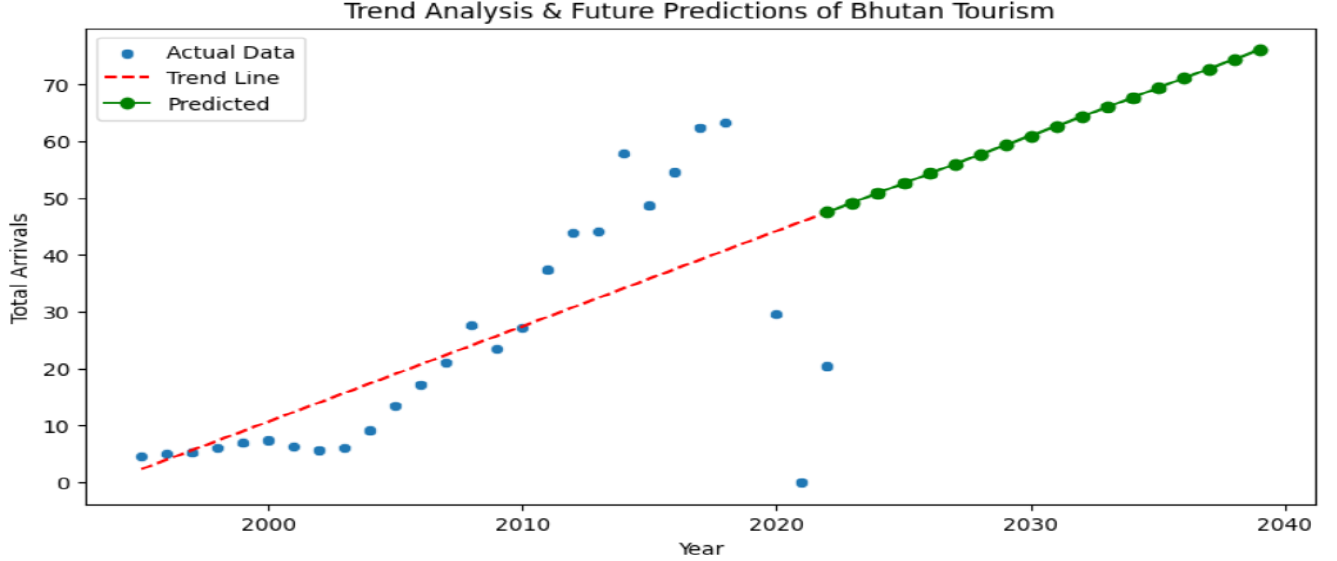


Figure 14. Trend Analysis and future predications of Tourism

Figure 14 illustrates a trend analysis and future predictions for tourism arrivals in Bhutan, using historical data and a predictive model. The training data relies on the single independent variable of "Year," which assumes a trend-based relationship between time and total arrivals.

Blue dots indicate actual recorded tourism arrivals, showing a general upward trend with fluctuations between 2010 and 2020.

The red dashed line represents the fitted linear trend, capturing the overall increase but failing to account for short-term variations or external factors.

Green points represent the model’s predictions from approximately 2025 to 2040, indicating a steady increase based on historical trends. Because the model uses only one variable, it doesn't account for factors like economic conditions, policies, travel restrictions, or global events, so the predictions may not fully reflect real-world complexities

# 7. Conclusion

This study used historical data and machine learning models, incorporating feature engineering techniques, to forecast tourism trends in Bhutan for 2025 and 2030. By assessing various predictive algorithms, it identified the most effective methods for predicting tourist arrivals.

The finding exposed patterns and insights into factors influencing Bhutan's tourism, such as global travel trends and its appeal as a peaceful destination. The analysis provided a detailed outlook on expected tourist arrivals and highlighted both challenges and opportunities for Bhutan's tourism sector.

# ANNEX

**#Step1**

!pip install beautifulsoup4 pandas numpy scikit-learn matplotlib seaborn streamlit plotly

**#Step2**

#Data Collection and Preprocessing

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

#Load dataset

BT = pd.read\_csv("BhutanTourismStatistics.csv")

# Display column names

print(BT.columns)

# Replace ".." or missing values with NaN

BT.replace("..", np.nan, inplace=True)

#print(BT)

#To replace nan with zero

BT.fillna(0, inplace=True)

#Check the dataset

print(BT.info())

# Convert object columns to numeric (if applicable)

BT = BT.apply(pd.to\_numeric, errors='coerce')

# Replace NaNs with 0 again (after conversion)

BT.fillna(0, inplace=True)

#Check the dataset

print(BT.info())

# summary statistics:

print(BT.describe())

# Check dataset

print(BT.head())

# Remove the 'Total Arrivals (000s)' and 'Same-day Visitors (000s)' columns since no value

BT.columns = BT.columns.str.strip()

BT = BT.drop(['Total Arrivals (000s)', 'Same-day Visitors (000s)'], axis=1)

print(BT.head())#Define region-based arrivals columns to have total arrivals

region\_columns = [

"Arrivals from Africa (000s)",

"Arrivals from America (000s)",

"Arrivals from East Asia and the Pacific (000s)",

"Arrivals from Europe (000s)",

"Arrivals from Middle East (000s)",

"Arrivals from South Asia (000s)"

]

# Create a new column for Total Arrivals by summing arrivals from all regions

BT ["Total Arrivals"] = BT[region\_columns].sum(axis=1)

**#Step3&4**

#Exploratory Data Analysis (EDA) & Feature Engineering

# Compute correlation matrix

plt.figure(figsize=(12,6))

sns.heatmap(BT.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=1)

plt.title("Feature Correlation Heatmap")

plt.show()

# Visualize Trends

#Total Arrivals Over Time (Line Chart)

plt.figure(figsize=(10,5))

sns.lineplot(x=BT["Year"], y=BT["Overnight Visitors (000s)"], marker="o", label="Total Arrivals")

plt.title("Total Arrivals Over Time")

plt.xlabel("Year")

plt.ylabel("Arrivals (000s)")

plt.grid(True)

plt.show()

#Arrivals Transport

BT\_transport = BT[["Year", "Arrivals by Air (000s)", "Arrivals by Land (000s)"]]

# Plot the data

BT\_transport.set\_index("Year").plot(kind="bar", stacked=True, figsize=(10, 6), colormap="viridis")

plt.title("Arrivals by Mode of Transport")

plt.xlabel("Year")

plt.ylabel("Arrivals (000s)")

plt.legend(title="Mode of Transport")

plt.show()

# Arrival for Business Vs Leisure

BT\_purpose = BT[["Year", "Arrivals for Leisure (000s)", "Arrivals for Business (000s)"]]

# Plot the data, ensuring all years are present on the x-axis

BT\_purpose.set\_index("Year").plot(kind="bar", figsize=(10, 6), colormap="coolwarm")

plt.title("Arrivals by Purpose")

plt.xlabel("Year")

plt.ylabel("Arrivals (000s)")

plt.legend(title="Purpose of Visit")

plt.show()

#Distribution of Expenditure

plt.figure(figsize=(8,4))

sns.histplot(BT["Total Expenditure (US$M)"], bins=15, kde=True)

plt.title("Distribution of Total Expenditure")

plt.xlabel("Expenditure (US$M)")

plt.show()

#Tourist Arrival by regions

# Plotting

plt.figure(figsize=(12, 6))

plt.plot(BT['Year'], BT['Arrivals from Africa (000s)'], label='Africa', marker='o')

plt.plot(BT['Year'], BT['Arrivals from America (000s)'], label='America', marker='o')

plt.plot(BT['Year'], BT['Arrivals from East Asia and the Pacific (000s)'], label='East Asia and the Pacific', marker='o')

plt.plot(BT['Year'], BT['Arrivals from Europe (000s)'], label='Europe', marker='o')

plt.plot(BT['Year'], BT['Arrivals from Middle East (000s)'], label='Middle East', marker='o')

plt.plot(BT['Year'], BT['Arrivals from South Asia (000s)'], label='South Asia', marker='o')

plt.title("Tourist Arrivals from different Regions (1995-2022)")

plt.xlabel("Year")

plt.ylabel("Arrivals (000s)")

plt.legend()

plt.grid(True)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.show()

# Remove outliers using IQR method

Q1 = BT["Total Arrivals"].quantile(0.25)

Q3 = BT["Total Arrivals"].quantile(0.75)

IQR = Q3 - Q1

BT = BT[(BT["Total Arrivals"] > (Q1 - 1.5 \* IQR)) & (BT["Total Arrivals"] < (Q3 + 1.5 \* IQR))]

#Creat New Features from Existing Data

#3-Year Moving Average of Total Arrivals

BT["MA\_Total\_Arrivals"] = BT["Total Arrivals"].rolling(window=3).mean()

#percentage change in total arrivals year over year

BT["Tourism\_Growth"] = BT["Total Arrivals"].pct\_change() \* 100

def create\_features(df, historical\_data=None):

""" Generate features and prevent NaN or inf values. """

df["Year\_Squared"] = df["Year"] \*\* 2 # Keep feature

if historical\_data is not None and "Total Arrivals" in historical\_data.columns:

# Compute rolling mean safely

df["MA\_Total\_Arrivals"] = historical\_data["Total Arrivals"].rolling(window=3, min\_periods=1).mean()

# Compute percentage change, handling division by zero

df["Tourism\_Growth"] = historical\_data["Total Arrivals"].pct\_change() \* 100

df["Tourism\_Growth"] = df["Tourism\_Growth"].replace([np.inf, -np.inf], np.nan) # Replace inf with NaN

df["Tourism\_Growth"] = df["Tourism\_Growth"].fillna(0) # Replace NaN with 0

# Instead of using inplace=True, use direct assignment

# Replace inf with NaN

# Replace NaN with 0

else:

df["MA\_Total\_Arrivals"] = 0

df["Tourism\_Growth"] = 0 # Default safe value

# Ensure all required columns exist

required\_features = [

"Overnight Visitors (000s)", "Arrivals by Air (000s)", "Arrivals by Land (000s)",

"Arrivals for Leisure (000s)", "Arrivals for Business (000s)", "Arrivals from Africa (000s)",

"Arrivals from America (000s)", "Arrivals from East Asia and the Pacific (000s)",

"Arrivals from Europe (000s)", "Arrivals from Middle East (000s)", "Arrivals from South Asia (000s)",

"Total Expenditure (US$M)", "Travel Expenditure (US$M)", "Passenger Transport (US$M)",

"Total Overnights (000s)", "Hotel Overnights (000s)", "MA\_Total\_Arrivals", "Tourism\_Growth"

]

for col in required\_features:

if col not in df.columns:

df[col] = historical\_data[col].mean() if historical\_data is not None else 0 # Use mean if available

# \*\*Remove any remaining inf or NaN values\*\*

df.replace([np.inf, -np.inf], np.nan, inplace=True)

df.fillna(0, inplace=True)

return df

# Plot Annual Growth Rate of Tourism

#shows how fast or slow tourism is growing

plt.figure(figsize=(10, 5))

sns.barplot(data=BT, x="Year", y="Tourism\_Growth", hue="Year", palette="coolwarm", legend=False)

plt.axhline(y=0, color='black', linestyle='--') # Reference line at 0% growth 20

plt.title("Annual Growth Rate of Tourism in Bhutan")

plt.xlabel("Year")

plt.ylabel("Growth Rate (%)")

plt.xticks(rotation=45)

plt.show()

# Plot 3-Year Moving Average of Total Arrivals

import matplotlib.pyplot as plt

import seaborn as sns

plt.figure(figsize=(10, 5))

sns.lineplot(data=BT, x="Year", y="Total Arrivals", label="Total Arrivals", marker="o")

sns.lineplot(data=BT, x="Year", y="MA\_Total\_Arrivals", label="3-Year Moving Average", marker="o", linestyle="--")

plt.title("3-Year Moving Average of Total Arrivals in Bhutan")

plt.xlabel("Year")

plt.ylabel("Total Arrivals (000s)")

plt.legend()

plt.show()

**# Step 5: Building and Training Machine Learning Models**

# Splitting the Data: Train & Test Sets

# Training Set (80%) – To train the model

# Test Set (20%) – To evaluate performance

from sklearn.model\_selection import train\_test\_split

# Define features (X) and target variable (y)

X = BT.drop(columns=["Year","Total Arrivals","Arrivals from Africa (000s)\_pct","Arrivals from America (000s)\_pct","Arrivals from East Asia and the Pacific (000s)\_pct","Arrivals from Europe (000s)\_pct",

"Arrivals from Middle East (000s)\_pct","Arrivals from South Asia (000s)\_pct","Arrivals from Africa (000s)\_growth","Arrivals from America (000s)\_growth","Arrivals from East Asia and the Pacific (000s)\_growth",

"Arrivals from Europe (000s) growth","Arrivals from Middle East (000s)\_growth","Arrivals from South Asia (000s)\_growth"]) # Drop non-predictive columns

y = BT["Total Arrivals"] # Target variable

# Split into training (80%) and testing (20%) sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**# Training Machine Learning Models**

**# Train a Linear Regression Model**

from sklearn.linear\_model import LinearRegression

**# Initialize and train the model**

lr\_model = LinearRegression()

lr\_model.fit(X\_train, y\_train)

**# Predict on test data**

y\_pred\_lr = lr\_model.predict(X\_test)

**#Train a Random Forest Model**

from sklearn.ensemble import RandomForestRegressor

# Initialize and train Random Forest model

rf\_model = RandomForestRegressor(n\_estimators=100, random\_state=42)

rf\_model.fit(X\_train, y\_train)

**# Predict on test data**

y\_pred\_rf = rf\_model.predict(X\_test)

**# Train an XGBoost Model 21**

from xgboost import XGBRegressor

**# Initialize and train XGBoost model**

xgb\_model = XGBRegressor(n\_estimators=100, learning\_rate=0.1, random\_state=42)

xgb\_model.fit(X\_train, y\_train)

# Predict on test data

y\_pred\_xgb = xgb\_model.predict(X\_test)

# All in One

**#Step 6: Evaluating Model Performance**

#Model Evaluation: Compare Performance

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import numpy as np

**# Function to evaluate models**

def evaluate\_model(model\_name, y\_test, y\_pred):

print(f" {model\_name} Model Performance:")

print(f" MAE: {mean\_absolute\_error(y\_test, y\_pred):.2f}")

print(f" MAE: {mean\_squared\_error(y\_test, y\_pred):.2f}")

print(f" RMSE: {np.sqrt(mean\_squared\_error(y\_test, y\_pred)):.2f}")

print(f" R² Score: {r2\_score(y\_test, y\_pred):.2f}\n")

**# Evaluate all models**

evaluate\_model("Linear Regression", y\_test, y\_pred\_lr)

evaluate\_model("Random Forest", y\_test, y\_pred\_rf)

evaluate\_model("XGBoost", y\_test, y\_pred\_xgb)

**# Step 1: Create future year DataFrame**

future\_years = pd.DataFrame({"Year": [2025, 2030]})

**# Step 2: Apply fixed feature engineering**

future\_features = create\_features(future\_years, historical\_data=BT)

**# Step 3: Ensure feature order matches training data**

future\_features = future\_features[X\_train.columns]

**# Step 4: \*\*Verify that `inf` is removed\*\***

print(future\_features.isin([np.nan, np.inf, -np.inf]).sum()) # Should print all zeros

**# Step 5: Make predictions using the trained model**

future\_predictions = lr\_model.predict(future\_features)

**# Step 6: Display results**

future\_years["Predicted\_Arrivals"] = future\_predictions

print(future\_years)

import matplotlib.pyplot as plt

import seaborn as sns

**# Plot Historical Data**

plt.figure(figsize=(10, 5))

sns.lineplot(x=BT["Year"], y=BT["Total Arrivals"], marker="o", label="Actual Data")

**# Plot Future Predictions**

sns.lineplot(x=future\_years["Year"], y=future\_years["Predicted\_Arrivals"], marker="o", linestyle="dashed", label="Predicted Data")

**# Formatting 22**

plt.xlabel("Year")

plt.ylabel("Total Arrivals")

plt.title("Tourism Arrivals Prediction Using Linear regression")

plt.legend()

plt.grid(True)

plt.show()

**# Scatter plot**

sns.regplot(x=BT["Year"], y=BT["Total Arrivals"], scatter\_kws={"color": "blue"}, line\_kws={"color": "red"}, label="Trend")

plt.scatter(future\_years["Year"], future\_years["Predicted\_Arrivals"], color="green", label="Future Predictions")

plt.xlabel("Year")

plt.ylabel("Total Arrivals")

plt.legend()

plt.title("Historical vs. Predicted Tourism Trends Using Linear regression")

plt.show()