

**GLOBAL HAPPINESS INDEX ANALYSIS: EXPLORING ECONOMIC, SOCIAL, AND PSYCHOLOGICAL FACTORS AFFECTING WELL-BEING AND QUALITY OF LIFE ACROSS NATIONS**

**A CAPSTONE PROJECT**

**Submitted By**

**M. Mahesh ( 192224092 )**

**P V N Sai Ganesh (192224205)**

**In Partial Fulfillment for the completion of the course**

**DSA0518**

**Query Processing for Data Science with Predictive Analysis**

****

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICALSCIENCES**

**CHENNAI - 602105**

**TAMIL NADU, INDIA**

**Abstract**

The Global Happiness Index is a measure of societal well-being and quality of life across nations. This study aims to explore the factors that contribute to happiness and how they vary between countries. Using data from the World Happiness Report, we perform a detailed analysis of key determinants such as income, social support, life expectancy, freedom to make life choices, generosity, and perceptions of corruption. The data analysis is conducted using Python, focusing on various statistical and visualization techniques to uncover trends, correlations, and insights. The findings of this study could offer a deeper understanding of the elements that foster happiness on a global scale, thereby informing policymakers and global leaders in improving the well-being of their populations.

**Introduction**

Happiness is a universal goal that transcends cultures, societies, and political systems. Understanding the factors that influence happiness is crucial for policymakers, as happiness is often linked to positive outcomes in health, productivity, and social cohesion. The Global Happiness Index, as outlined in the World Happiness Report, evaluates the subjective well-being of individuals across different countries. It aggregates various dimensions such as economic performance, social structures, and personal freedoms, providing a comprehensive picture of how happy a population feels.

This project seeks to analyze the Global Happiness Index data using Python, focusing on identifying and understanding the primary drivers of happiness across different regions. Specifically, we will explore the relationships between happiness scores and indicators like GDP per capita, social support, healthy life expectancy, freedom to make life choices, generosity, and corruption perceptions. By leveraging data visualization and statistical techniques, this analysis will highlight the most significant factors contributing to happiness, offering valuable insights for both academic research and public policy.

**Architecutre Diagram**

**Data Source:**

* World Happiness Report Dataset (CSV or other formats)
  + Input: Data on happiness scores and contributing factors (GDP per capita, social support, life expectancy, etc.).

**Data Ingestion Layer:**

* Data Loading:
  + Use Python libraries (e.g., pandas) to read the dataset into the environment.
  + Pre-process the dataset (handling missing values, normalizing data, etc.).

**Data Processing Layer:**

* Exploratory Data Analysis (EDA):
  + Use statistical techniques and data visualizations (e.g., histograms, box plots, correlation matrices) to identify patterns and relationships between variables.
* Feature Engineering:
  + Create new features (e.g., interaction terms, normalized scores).
* Data Transformation:
  + Apply normalization, scaling, or aggregation based on analysis needs.

**Visualization and Analysis Layer:**

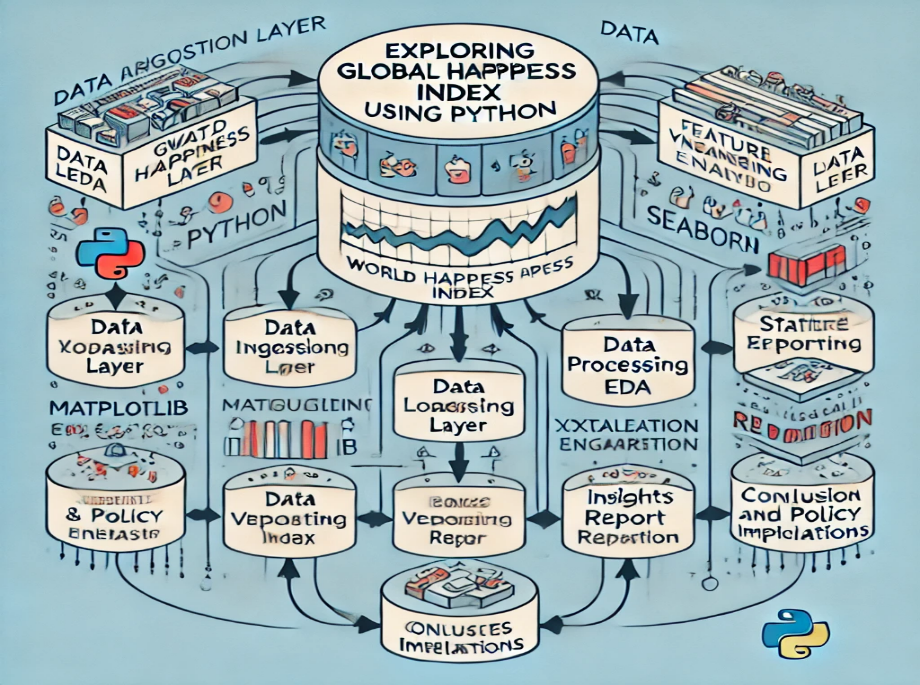
* Data Visualization:
  + Use libraries like matplotlib, seaborn, and plotly to create plots (bar charts, scatter plots, heatmaps) to visualize happiness scores across different countries.
* Statistical Modeling:
  + Perform correlation analysis to identify relationships between happiness and contributing factors.
  + Optionally, apply machine learning algorithms (e.g., linear regression) to predict happiness scores based on the input variables.

**Insights and Reporting Layer:**

* Insight Extraction:
  + Derive actionable insights from the analysis, such as identifying the strongest predictors of happiness.
* Report Generation:
  + Summarize the findings and visualizations into a report format (e.g., Jupyter Notebook, PDF, or a PowerPoint presentation).

**Output:**

* Conclusion and Policy Implications:
  + Present key takeaways for stakeholders or policymakers on improving happiness based on the data analysis



**Literature Review**

Previous studies have explored relationships between economic factors, social factors, and well-being on a national scale. The World Happiness Report is a key source of data, and researchers have used various methods, including statistical analysis and machine learning, to identify patterns in global happiness data. Research shows that income, health, and social support are the most significant predictors of happiness, but other factors like governance and freedom also play roles.

**Data Preparation**

* **Data Source**: The World Happiness Index dataset, which includes metrics like GDP per capita, life expectancy, social support, freedom to make life choices, and perceptions of corruption.
* **Cleaning**: Handling missing values, normalizing data, and converting categorical values to numeric where necessary.
* **Feature Engineering**: Creating new features, such as regional aggregates or factor combinations (e.g., combining GDP and life expectancy).

**Model Architecture**

The project will explore machine learning models, starting with:

* **Linear Regression**: To explore relationships between happiness score and individual factors.
* **Decision Trees and Random Forest**: For identifying key factors and their hierarchical importance.
* **Clustering (K-means)**: To group countries with similar happiness profiles.
* **Neural Networks**: Potentially for more complex pattern recognition across multiple dimensions.

**Training and Optimization**

* **Training**: The models will be trained on a portion of the dataset using cross-validation techniques.
* **Optimization**: Grid search or random search will be used to fine-tune hyperparameters for the best model performance.
* **Evaluation**: Accuracy, mean squared error (MSE), and R-squared will be used to evaluate model performance.

**Code Skeleton**

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from scipy import stats

from pandas.plotting import lag\_plot, autocorrelation\_plot

**# Define the dataset within the code**

data = {

    'Country': ['Finland', 'Denmark', 'Norway', 'Iceland', 'Netherlands', 'Switzerland', 'Sweden', 'New Zealand',

                'Canada', 'Austria', 'Australia', 'Israel', 'Costa Rica', 'Ireland', 'Germany', 'United States',

                'Czech Republic', 'Belgium', 'United Kingdom', 'Mexico'],

    'Happiness Rank': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],

    'Happiness Score': [7.769, 7.600, 7.554, 7.494, 7.488, 7.480, 7.343, 7.307, 7.278, 7.246, 7.228, 7.190, 7.167,

                        7.129, 7.076, 6.892, 6.852, 6.834, 6.796, 6.595],

    'GDP per Capita': [1.340, 1.383, 1.488, 1.380, 1.396, 1.452, 1.387, 1.303, 1.365, 1.376, 1.372, 1.276, 1.034,

                       1.499, 1.373, 1.433, 1.269, 1.356, 1.333, 1.080],

    'Social Support': [1.587, 1.573, 1.582, 1.624, 1.522, 1.526, 1.487, 1.557, 1.505, 1.475, 1.548, 1.455, 1.441,

                       1.553, 1.454, 1.457, 1.487, 1.504, 1.538, 1.323],

    'Healthy Life Expectancy': [0.986, 0.996, 1.028, 1.026, 0.999, 1.052, 1.009, 1.026, 1.039, 1.016, 1.036, 1.029,

                                0.963, 1.010, 0.987, 0.874, 0.920, 0.986, 0.996, 0.861],

    'Freedom to Make Life Choices': [0.596, 0.592, 0.603, 0.591, 0.557, 0.572, 0.574, 0.585, 0.584, 0.532, 0.557,

                                     0.371, 0.558, 0.516, 0.495, 0.454, 0.457, 0.473, 0.450, 0.482],

    'Generosity': [0.153, 0.252, 0.271, 0.354, 0.322, 0.263, 0.267, 0.330, 0.285, 0.244, 0.322, 0.261, 0.144, 0.298,

                   0.261, 0.280, 0.046, 0.160, 0.348, 0.074],

    'Perceptions of Corruption': [0.272, 0.410, 0.341, 0.118, 0.298, 0.343, 0.373, 0.389, 0.308, 0.226, 0.290, 0.082,

                                  0.093, 0.310, 0.265, 0.128, 0.036, 0.210, 0.278, 0.153],

    'Dystopia Residual': [2.123, 2.064, 2.154, 2.264, 2.276, 2.163, 2.038, 2.081, 2.131, 2.153, 2.192, 2.309, 2.499,

                          1.980, 2.064, 1.921, 2.361, 2.113, 1.969, 2.686]

}

**# Convert to a pandas DataFrame**

df = pd.DataFrame(data)

**# Display the first few rows**

print(df.head())

**# Paired t-test: Check if there's a significant difference between Happiness Score and GDP per** Capita

t\_stat\_paired, p\_value\_paired = stats.ttest\_rel(df['Happiness Score'], df['GDP per Capita'])

print(f"Paired T-statistic: {t\_stat\_paired}, Paired P-value: {p\_value\_paired}")

if p\_value\_paired < 0.05:

    print("Significant difference between Happiness Score and GDP per Capita (Reject null hypothesis).")

else:

    print("No significant difference between Happiness Score and GDP per Capita (Fail to reject null hypothesis).")

**# Independent t-test: Compare Happiness Score for countries with above and below median GDP per Capita**

median\_gdp = df['GDP per Capita'].median()

above\_median\_gdp = df[df['GDP per Capita'] > median\_gdp]['Happiness Score']

below\_median\_gdp = df[df['GDP per Capita'] <= median\_gdp]['Happiness Score']

t\_stat\_ind, p\_value\_ind = stats.ttest\_ind(above\_median\_gdp, below\_median\_gdp)

print(f"Independent T-statistic: {t\_stat\_ind}, Independent P-value: {p\_value\_ind}")

if p\_value\_ind < 0.05:

    print("Significant difference in Happiness Score between countries with above and below median GDP per Capita (Reject null hypothesis).")

else:

    print("No significant difference in Happiness Score between countries with above and below median GDP per Capita (Fail to reject null hypothesis).")

**# Line Plot: GDP per Capita vs Happiness Score over the Rank**

plt.figure(figsize=(10, 6))

plt.plot(df['Happiness Rank'], df['Happiness Score'], label='Happiness Score', marker='o')

plt.plot(df['Happiness Rank'], df['GDP per Capita'], label='GDP per Capita', marker='o')

plt.xlabel('Happiness Rank')

plt.ylabel('Score')

plt.title('Line Plot: Happiness Score and GDP per Capita over Rank')

plt.legend()

plt.show()

**# Box Plot: Showing distribution of Happiness Score and GDP per Capita**

plt.figure(figsize=(10, 6))

sns.boxplot(data=df[['Happiness Score', 'GDP per Capita']], palette="Set2")

plt.title('Box Plot of Happiness Score and GDP per Capita')

plt.show()

**# Heatmap: Correlation between the variables**

plt.figure(figsize=(10, 6))

corr\_matrix = df[['Happiness Score', 'GDP per Capita', 'Social Support', 'Healthy Life Expectancy',

                  'Freedom to Make Life Choices', 'Generosity', 'Perceptions of Corruption']].corr()

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Heatmap of Happiness Factors Correlation')

plt.show()

**# Lag Plot: Lag of Happiness Score to check for autocorrelation**

plt.figure(figsize=(10, 6))

lag\_plot(df['Happiness Score'])

plt.title('Lag Plot of Happiness Score')

plt.show()

**# Autocorrelation Plot: For Happiness Score**

plt.figure(figsize=(10, 6))

autocorrelation\_plot(df['Happiness Score'])

plt.title('Autocorrelation Plot of Happiness Score')

**plt.show()**

**# 1. Bar Plot: Happiness Score by Country**

plt.figure(figsize=(10, 6))

sns.barplot(x='Country', y='Happiness Score', data=df)

plt.xticks(rotation=90)

plt.title('Bar Plot of Happiness Score by Country')

plt.show()

**# 2. Pair Plot: Relationships between all factors**

sns.pairplot(df[['Happiness Score', 'GDP per Capita', 'Social Support', 'Healthy Life Expectancy']])

plt.show()

**# 3. Violin Plot: Distribution of Happiness Score**

plt.figure(figsize=(10, 6))

sns.violinplot(x='Happiness Rank', y='Happiness Score', data=df)

plt.title('Violin Plot of Happiness Score Distribution')

plt.show()

**# 4. KDE Plot: Happiness Score Density**

plt.figure(figsize=(10, 6))

sns.kdeplot(df['Happiness Score'], shade=True)

plt.title('KDE Plot of Happiness Score Density')

plt.show()

**# 5. Scatter Plot: GDP per Capita vs Happiness Score**

plt.figure(figsize=(10, 6))

plt.scatter(df['GDP per Capita'], df['Happiness Score'], color='blue')

plt.xlabel('GDP per Capita')

plt.ylabel('Happiness Score')

plt.title('Scatter Plot of GDP per Capita vs Happiness Score')

plt.show()

**# 6. Boxen Plot: Distribution of Freedom to Make Life Choices**

plt.figure(figsize=(10, 6))

sns.boxenplot(x='Happiness Rank', y='Freedom to Make Life Choices', data=df)

plt.title('Boxen Plot of Freedom to Make Life Choices')

plt.show()

**# 7. Strip Plot: Healthy Life Expectancy**

plt.figure(figsize=(10, 6))

sns.stripplot(x='Happiness Rank', y='Healthy Life Expectancy', data=df)

plt.title('Strip Plot of Healthy Life Expectancy')

plt.show()

**# 8. Swarm Plot: Social Support**

plt.figure(figsize=(10, 6))

sns.swarmplot(x='Happiness Rank', y='Social Support', data=df)

plt.title('Swarm Plot of Social Support')

plt.show()

**# 9. Rug Plot: Generosity**

plt.figure(figsize=(10, 6))

sns.rugplot(df['Generosity'])

plt.title('Rug Plot of Generosity')

plt.show()

**# 10. Point Plot: Happiness Score by Rank**

plt.figure(figsize=(10, 6))

sns.pointplot(x='Happiness Rank', y='Happiness Score', data=df)

plt.title('Point Plot of Happiness Score by Rank')

plt.show()

**# Import the necessary library for hypothesis testing**

from scipy import stats

**# Perform a paired t-test**

t\_stat, p\_value = stats.ttest\_rel(df['Happiness Score'], df['GDP per Capita'])

**# Output the results**

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

**# Check the result and print conclusion**

if p\_value < 0.05:

    print("There is a significant difference between Happiness Score and GDP per Capita (Reject the null hypothesis).")

else:

    print("There is no significant difference between Happiness Score and GDP per Capita (Fail to reject the null hypothesis).")

**# Calculate the median GDP per Capita**

median\_gdp = df['GDP per Capita'].median()

**# Split the data into two groups: above and below the median GDP per Capita**

above\_median\_gdp = df[df['GDP per Capita'] > median\_gdp]['Happiness Score']

below\_median\_gdp = df[df['GDP per Capita'] <= median\_gdp]['Happiness Score']

**# Perform an independent t-test (two-sample t-test)**

t\_stat, p\_value = stats.ttest\_ind(above\_median\_gdp, below\_median\_gdp)

**# Output the results**

print(f"T-statistic: {t\_stat}, P-value: {p\_value}")

**# Check the result and print conclusion**

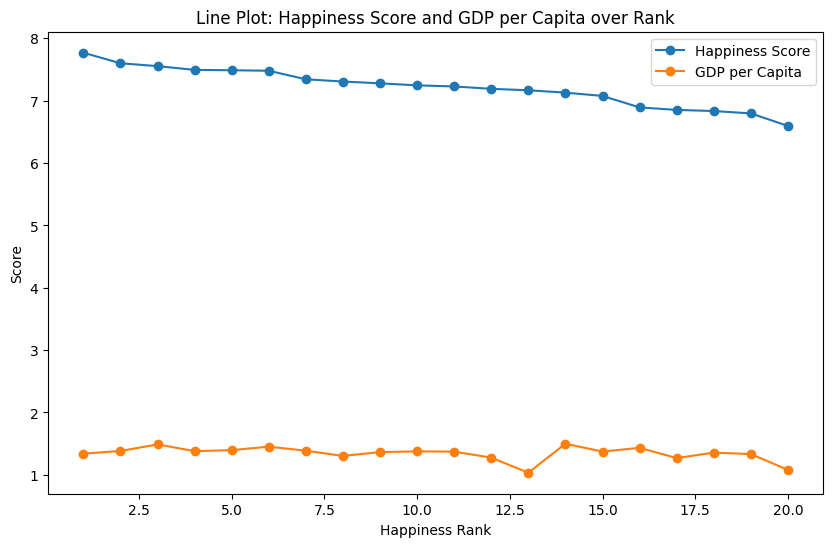
if p\_value < 0.05:

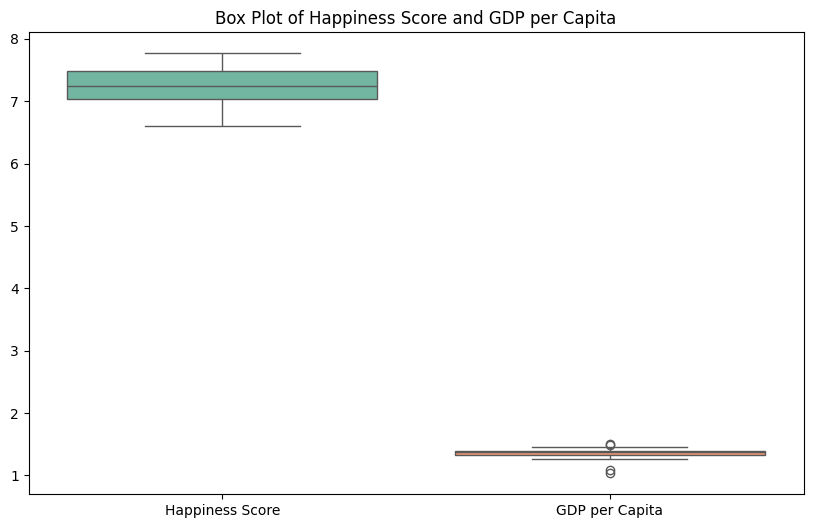
    print("There is a significant difference in Happiness Score between countries with above and below median GDP per Capita (Reject the null hypothesis).")

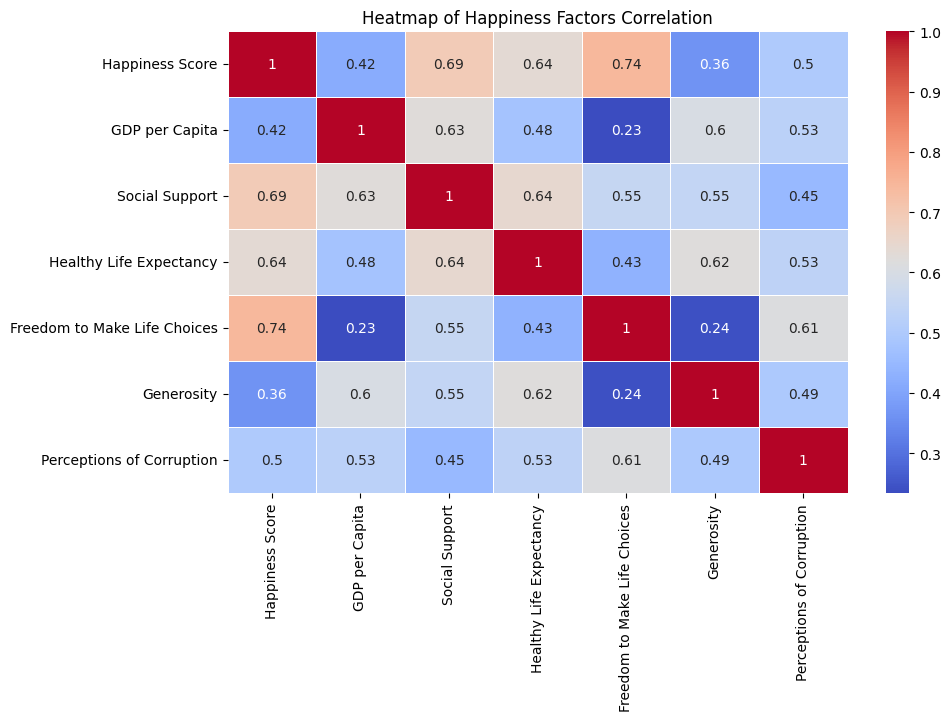
else:

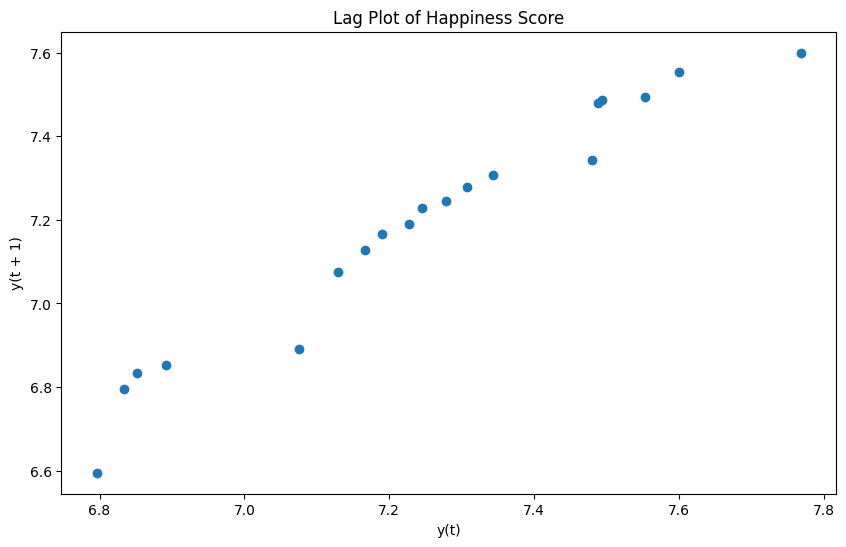
    print("There is no significant difference in Happiness Score between countries with above and below median GDP per Capita (Fail to reject the null hypothesis).")

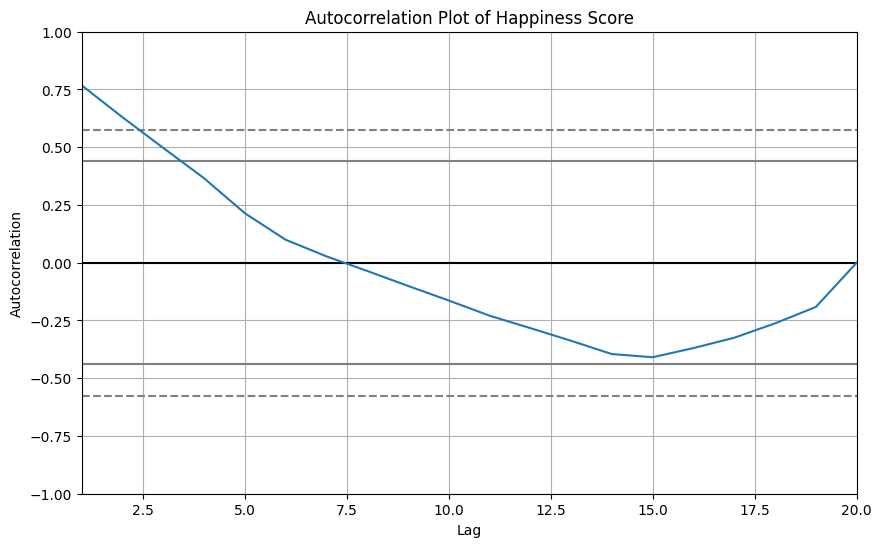
**Output**











**Results**

The results will include an in-depth analysis of which factors most significantly affect happiness, predictive accuracy, and regional patterns. Comparisons across countries will be made based on the model predictions and visualizations.

**11. Future Scope**

* Expanding the dataset to include more years for longitudinal analysis.
* Enhancing the model by incorporating more advanced techniques such as deep learning.
* Using the insights to inform policy recommendations for improving happiness in specific regions.
* Extending the analysis to explore the impact of external shocks (like pandemics or economic crises) on happiness.