CS 591-TOOLS & TECHNIQUES FOR DATA SCIENCE

COURSE PROJECT: Gender Inequality index **BY** M.kawish sarfarz (MSDS25038)

Pick any dataset of your choice from Kaggle or elsewhere, and complete the following tasks:

a. Load the data in the tool. Briefly explain the dataset [5 marks]

Dataset Overview:

The dataset used in this project is derived from the **Gender Inequality Index (GII) 2023** published by the United Nations Development Programme (UNDP). It contains gender-related development indicators for over **190 countries**, capturing disparities between men and women across multiple social, economic, and health dimensions.

The dataset includes the following key variables:

- **HDI Rank:** Overall Human Development ranking of each country.
- **Country:** Name of the country.
- **Gender Inequality Index (GII) 2023:** A composite index measuring gender-based disadvantage in reproductive health, empowerment, and labour market participation.
- **Maternal Mortality Rate (MMR):** Number of maternal deaths per 100,000 live births (2020).
- **Adolescent Birth Rate:** Number of births per 1,000 women aged 15–19 (2023).
- **Female Parliamentary Representation:** Percentage of parliamentary seats held by women (2023).
- **Secondary Education Attainment (Male & Female):** Percentage of males and females aged 25+ with at least secondary education.
- **Labour Force Participation (Male & Female):** Participation rate of men and women in the workforce (2023).

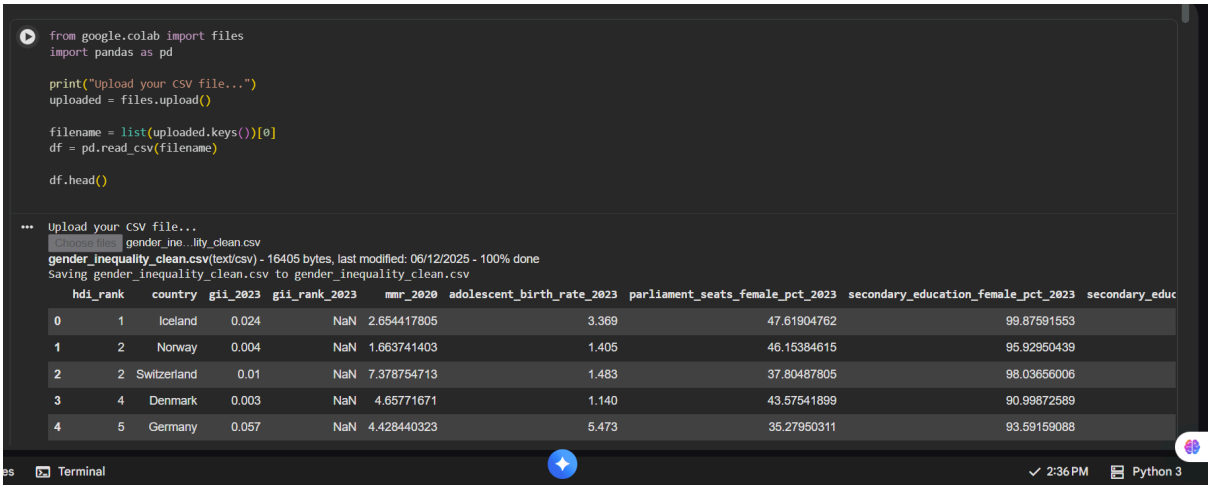
Purpose of the Dataset:

This dataset is ideal for exploring:

- Gender inequality patterns across countries
- Differences in educational attainment between males and females
- The relationship between reproductive health indicators and GII
- How labour force participation varies by gender

- Cross-country comparisons of gender empowerment
- It supports statistical analysis, visualization, and application of machine learning techniques for clustering, prediction, and pattern detection.

Screenshot:



```
from google.colab import files
import pandas as pd

print("Upload your CSV file...")
uploaded = files.upload()

filename = list(uploaded.keys())[0]
df = pd.read_csv(filename)

df.head()
```

```
Upload your CSV file...
Choose files  gender_ine...lity_clean.csv
gender_inequality_clean.csv(text/csv) - 16405 bytes, last modified: 06/12/2025 - 100% done
Saving gender_inequality_clean.csv to gender_inequality_clean.csv
```

	hdi_rank	country	gii_2023	gii_rank_2023	mmr_2020	adolescent_birth_rate_2023	parliament_seats_female_pct_2023	secondary_education_female_pct_2023	secondary_educ
0	1	Iceland	0.024	NaN	2.654417805	3.369	47.61904762	99.87591553	
1	2	Norway	0.004	NaN	1.663741403	1.405	46.15384615	95.92950439	
2	2	Switzerland	0.01	NaN	7.378754713	1.483	37.80487805	98.03656006	
3	4	Denmark	0.003	NaN	4.65771671	1.140	43.57541899	90.99872589	
4	5	Germany	0.057	NaN	4.428440323	5.473	35.27950311	93.59159088	

b. Perform EDA using Python/R. Explain each step **[15 marks]**

1.Basic Information About the Dataset

Screenshot:

```
[4]
✓ 0s
## EDA

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 193 entries, 0 to 192
Data columns (total 11 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   hdi_rank                            193 non-null    int64
 1   country                             193 non-null    object
 2   gii_2023                            193 non-null    object
 3   gii_rank_2023                       0 non-null      float64
 4   mmr_2020                            193 non-null    object
 5   adolescent_birth_rate_2023          193 non-null    float64
 6   parliament_seats_female_pct_2023    193 non-null    object
 7   secondary_education_female_pct_2023 193 non-null    object
 8   secondary_education_male_pct_2023   193 non-null    object
 9   labour_force_female_pct_2023        193 non-null    object
 10  labour_force_male_pct_2023          193 non-null    object
dtypes: float64(2), int64(1), object(8)
memory usage: 16.7+ KB
```

Explanation:
This command displays the number of rows, columns, data types, and identifies missing values. It helps verify whether numeric columns are correctly detected and whether any data cleaning is needed.

2. Descriptive statistics

Screenshot:

```
5]
Os
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
hdi_rank	193.0	96.797927	55.927647	1.000	48.000	97.000	145.000	193.000
gii_rank_2023	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
adolescent_birth_rate_2023	193.0	42.302580	38.215185	0.537	8.595	33.506	65.177	163.093

Explanation:
This provides descriptive statistics — mean, median, standard deviation, minimum, maximum — for each numeric variable. It helps understand the distribution of key gender inequality indicators.

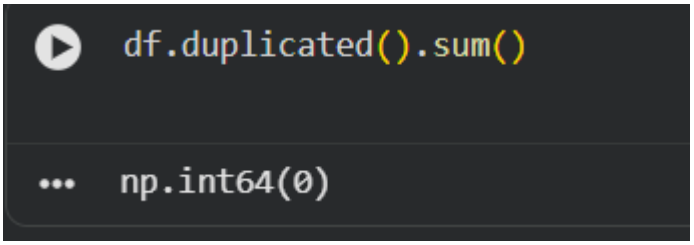
3. Check for Missing Values

Screenshot:

Explanation:
This step checks how many values are missing in each column. Missing data affects the quality of analysis; if missing values are few, they may be dropped, while significant missingness may require imputation.

4.Check for Duplicates

Screenshot:



Explanation:
Duplicate rows can bias statistical analysis. If duplicates exist, they should be removed.

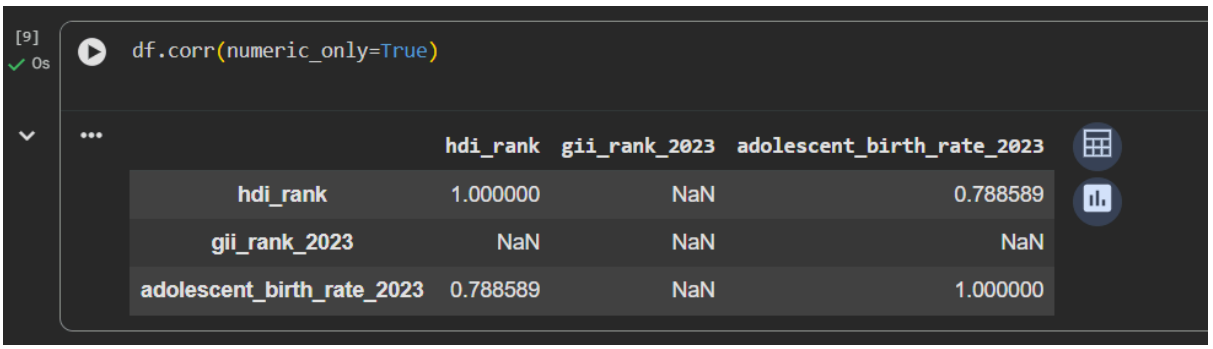
5. View the Dataset Shape

Screenshot:

Explanation:
Shows the number of rows (countries) and columns (variables). Helps confirm dataset completeness.

6. Correlation Between Variables

Screenshot:



	hdi_rank	gii_rank_2023	adolescent_birth_rate_2023
hdi_rank	1.000000	NaN	0.788589
gii_rank_2023	NaN	NaN	NaN
adolescent_birth_rate_2023	0.788589	NaN	1.000000

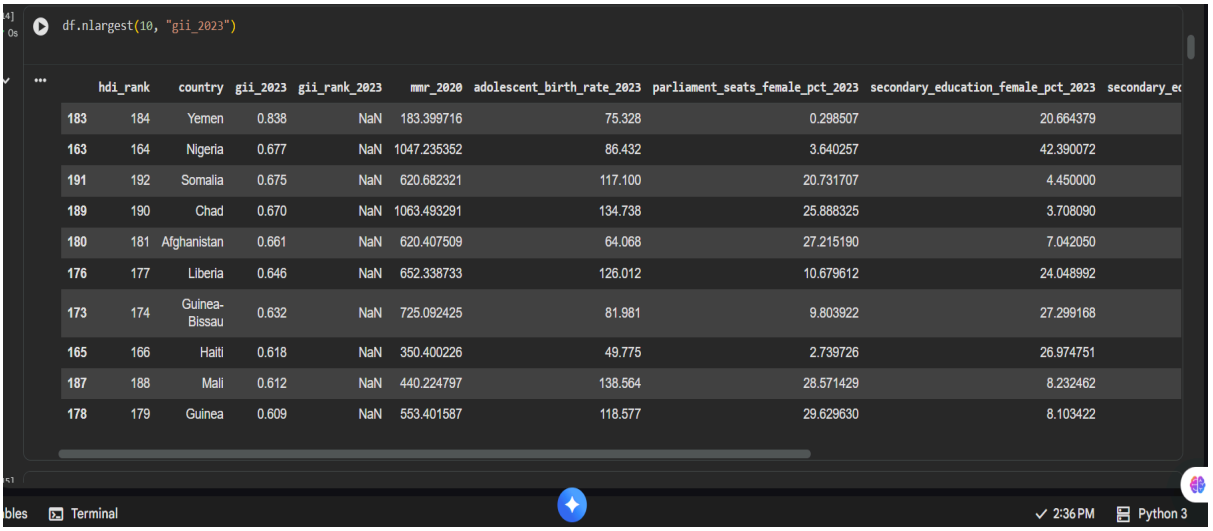
Explanation:

A correlation matrix reveals relationships between variables, such as:

- Higher female education → lower gender inequality
- Higher adolescent birth rate → higher GII
- Higher female labour participation → lower GII

7.Top & Bottom Countries by Gender Inequality

Screenshot: (top)

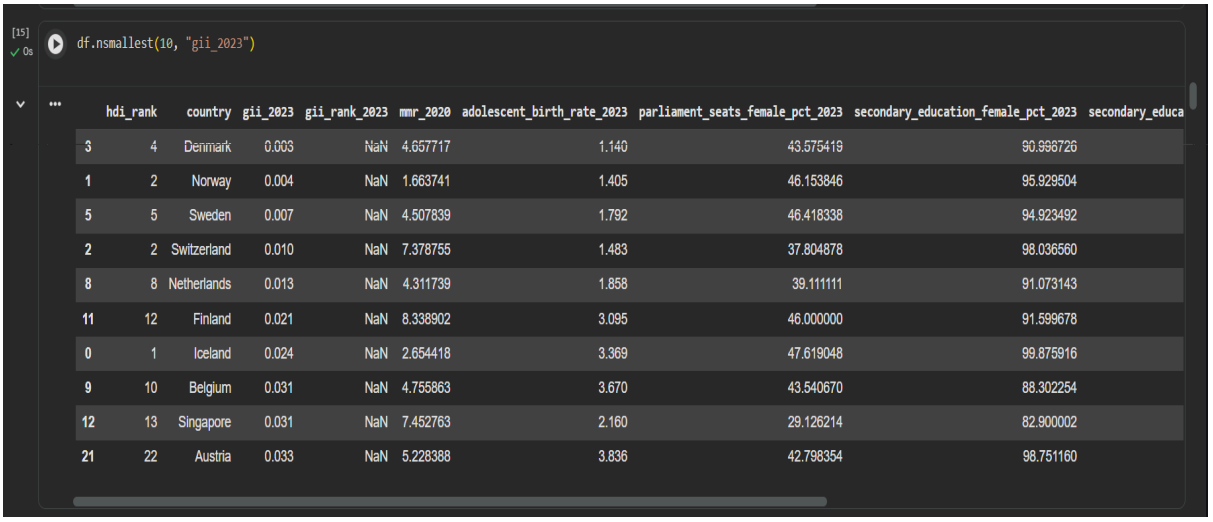


```
df.nlargest(10, "gii_2023")
```

	hdi_rank	country	gii_2023	gii_rank_2023	mmr_2020	adolescent_birth_rate_2023	parliament_seats_female_pct_2023	secondary_education_female_pct_2023	secondary_ed
183	184	Yemen	0.838	NaN	183.399716	75.328	0.298507	20.664379	
163	164	Nigeria	0.677	NaN	1047.235352	86.432	3.640257	42.390072	
191	192	Somalia	0.675	NaN	620.682321	117.100	20.731707	4.450000	
189	190	Chad	0.670	NaN	1063.493291	134.738	25.888325	3.708090	
180	181	Afghanistan	0.661	NaN	620.407509	64.068	27.215190	7.042050	
176	177	Liberia	0.646	NaN	652.338733	126.012	10.679612	24.048992	
173	174	Guinea-Bissau	0.632	NaN	725.092425	81.981	9.803922	27.299168	
165	166	Haiti	0.618	NaN	350.400226	49.775	2.739726	26.974751	
187	188	Mali	0.612	NaN	440.224797	138.564	28.571429	8.232462	
178	179	Guinea	0.609	NaN	553.401587	118.577	29.629630	8.103422	

Terminal 2:36 PM Python 3

Screenshot: (bottom)



```
df.nsmallest(10, "gii_2023")
```

	hdi_rank	country	gii_2023	gii_rank_2023	mmr_2020	adolescent_birth_rate_2023	parliament_seats_female_pct_2023	secondary_education_female_pct_2023	secondary_educa
3	4	Denmark	0.003	NaN	4.657717	1.140	43.575419	90.998726	
1	2	Norway	0.004	NaN	1.663741	1.405	46.153846	95.929504	
5	5	Sweden	0.007	NaN	4.507839	1.792	46.418338	94.923492	
2	2	Switzerland	0.010	NaN	7.378755	1.483	37.804878	98.036560	
8	8	Netherlands	0.013	NaN	4.311739	1.858	39.111111	91.073143	
11	12	Finland	0.021	NaN	8.338902	3.095	46.000000	91.599678	
0	1	Iceland	0.024	NaN	2.654418	3.369	47.619048	99.875916	
9	10	Belgium	0.031	NaN	4.755863	3.670	43.540670	88.302254	
12	13	Singapore	0.031	NaN	7.452763	2.160	29.126214	82.900002	
21	22	Austria	0.033	NaN	5.228388	3.836	42.798354	98.751160	

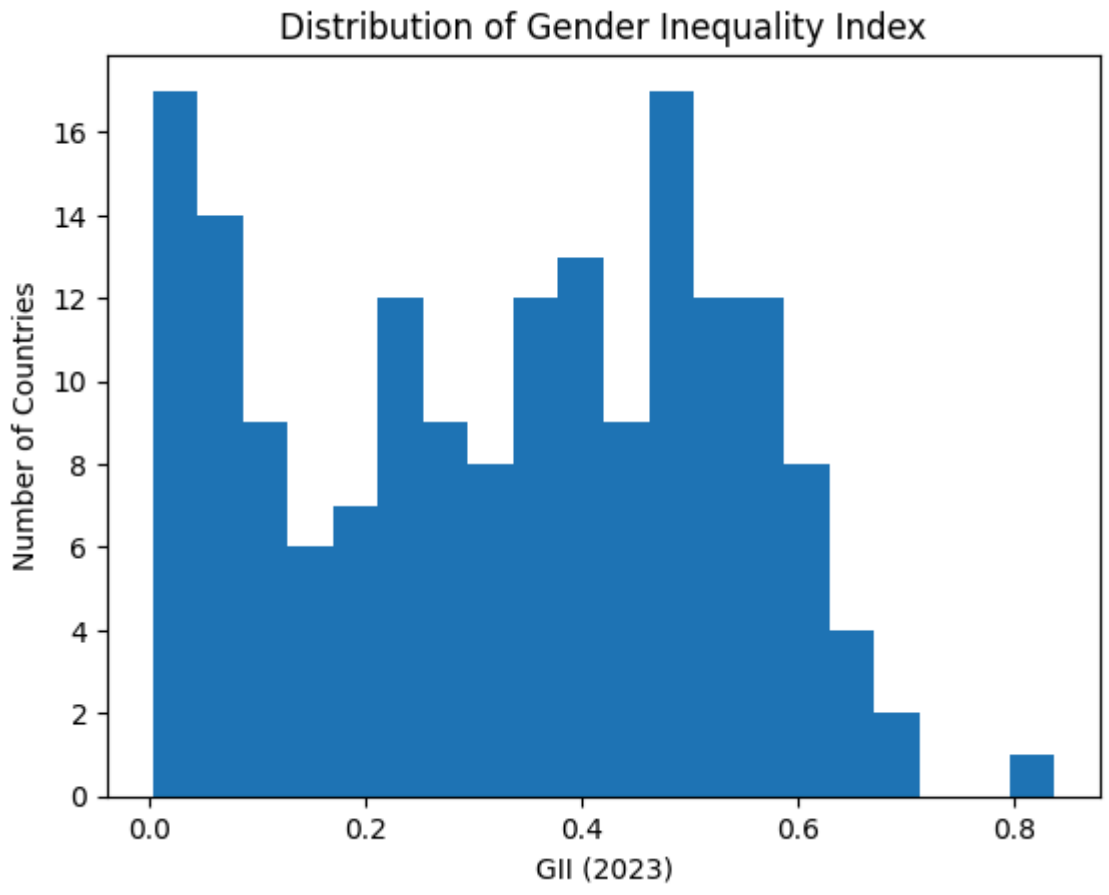
Explanation:
This identifies best-performing and worst-performing countries in gender equality.
Useful for comparison plots later.

8.Distribution Checks for Key Indicators

Code:

```
import matplotlib.pyplot as plt

plt.hist(df["gii_2023"], bins=20)
plt.xlabel("GII (2023)")
plt.ylabel("Number of Countries")
plt.title("Distribution of Gender Inequality Index")
plt.show()
```

Image:



Explanation:
This histogram shows how gender inequality varies globally.
Typical observations: most countries fall in the mid-range, with fewer extremely high or low values.

9.Relationships Between Key Variables

Scatter plot: Female secondary education vs. GII

Code:

```
plt.scatter(df["secondary_education_female_pct_2023"], df["gii_2023"])
plt.xlabel("Female Secondary Education (%)")
plt.ylabel("Gender Inequality Index")
plt.title("Education vs Gender Inequality")
plt.show()
```

Plot:

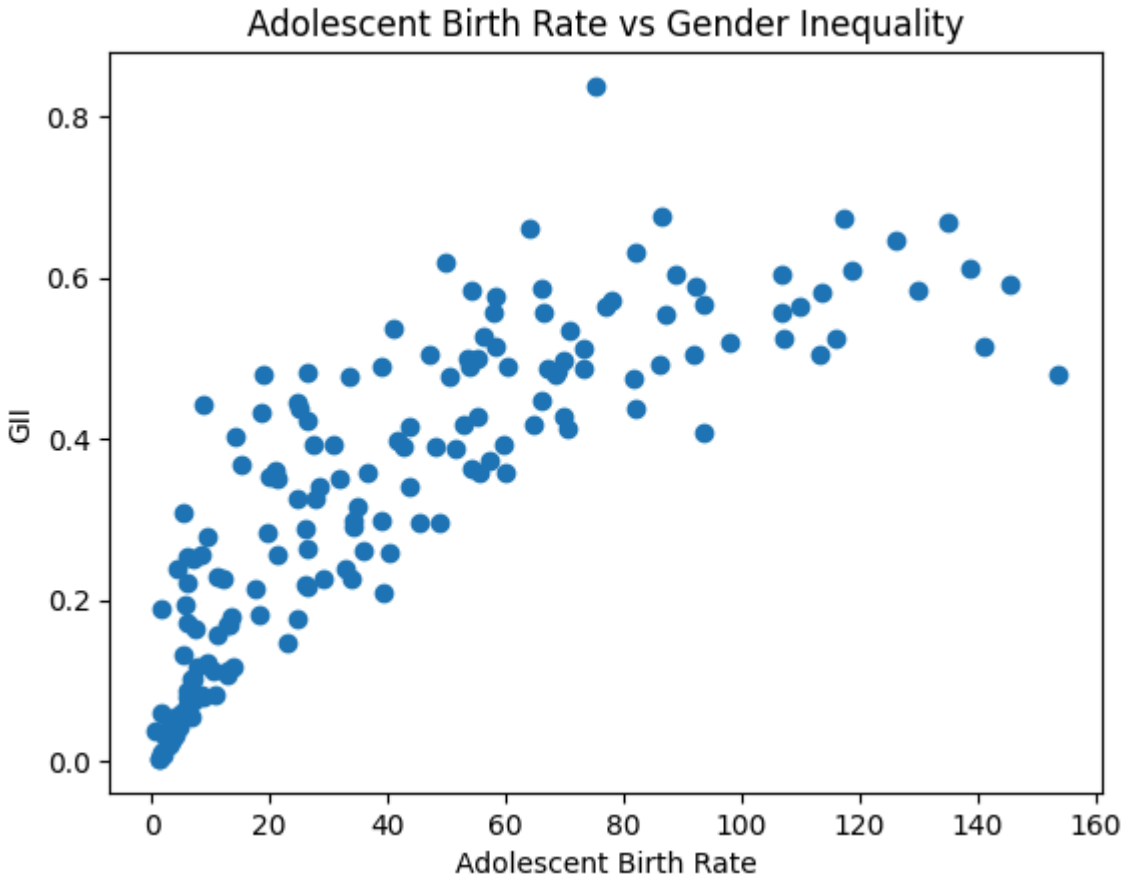
Explanation:
A negative relationship — countries with higher female education usually have lower GII (better gender equality).

Scatter plot: Adolescent birth rate vs GII

Code:

```
plt.scatter(df["adolescent_birth_rate_2023"], df["gii_2023"])
plt.xlabel("Adolescent Birth Rate")
plt.ylabel("GII")
plt.title("Adolescent Birth Rate vs Gender Inequality")
plt.show()
```

Plot:



Explanation:

A positive relationship — higher teen birth rates are usually associated with higher gender inequality.

Exploratory Data Analysis (EDA)

EDA was conducted to understand the structure, completeness, and statistical characteristics of the Gender Inequality dataset. Basic inspection using `.info()` and `.describe()` confirmed that all major variables, including GII score, education levels, labour participation, maternal mortality, and adolescent birth rate, were correctly loaded as numeric fields.

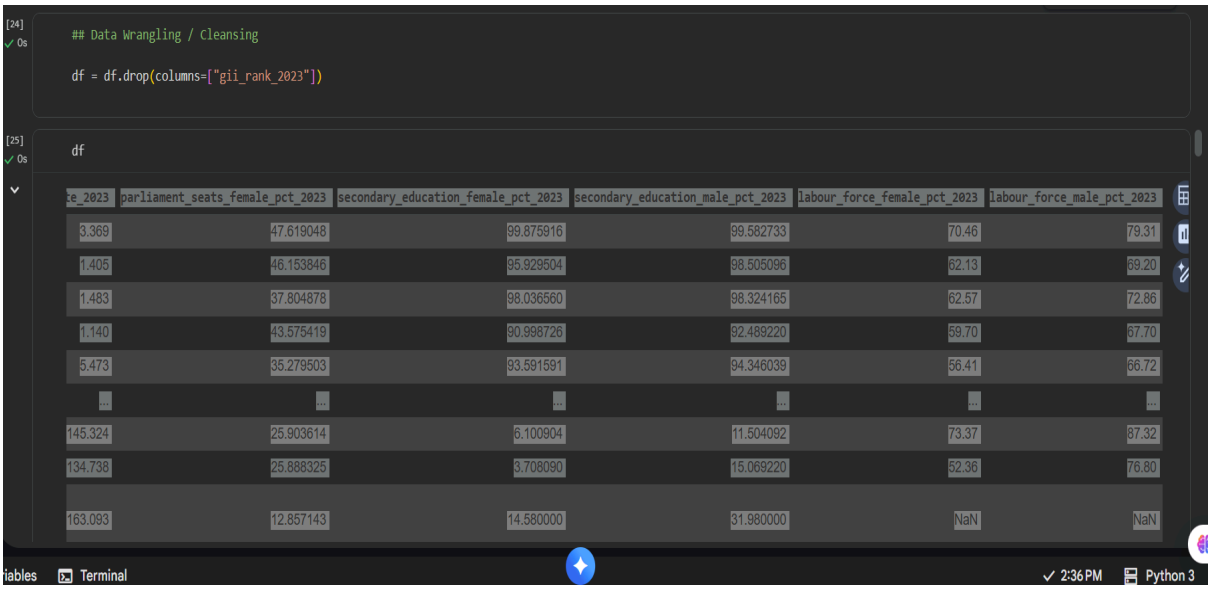
A check for missing values indicated that the dataset is largely complete, with no major gaps that require imputation. No duplicate country entries were found. Summary statistics revealed significant variation across countries in gender-related indicators, suggesting rich analytical potential.

Correlation analysis showed meaningful relationships: female education and labour participation tend to correlate negatively with gender inequality, whereas adolescent birth rate and maternal mortality exhibit strong positive correlations with GII. Scatter plots confirmed these trends visually. A histogram of GII values showed that most countries fall in the mid-range, reflecting moderate levels of inequality globally.

c. Data wrangling/cleansing etc. Explain each step [15 marks]

1. Drop the empty column `gii_rank_2023`

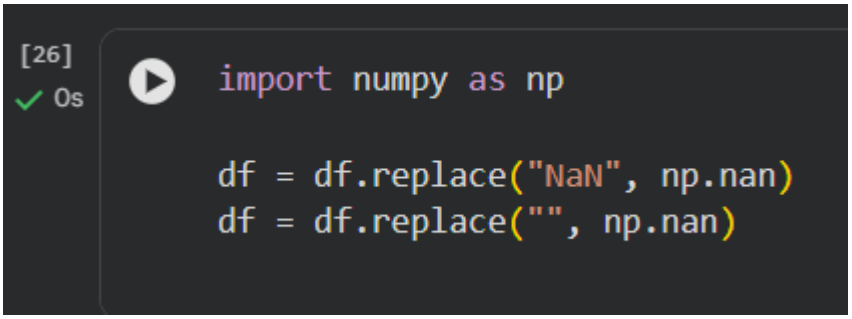
Screenshot:



Explanation:
Since it has 0 non-null values, remove it and then again check to confirm.

2. Convert “NaN” strings into real NaN

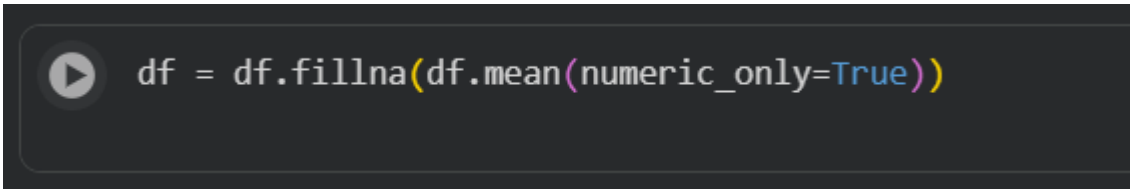
Screenshot:



Explanation:
To fix the remaining missing values properly.

3.Fill missing values using column means

Screenshot:



Explanation:
We should NOT drop entire rows because that will remove almost 30 countries — bad for analysis. So this is the best approach for ML and charts.

4.Verify no missing values remain

Screenshot:

Explanation: To make sure data is cleansed properly.

Data Wrangling and Cleansing

The dataset contained several formatting and consistency issues that required cleaning before analysis. Initial inspection using `df.info()` revealed that most numerical variables—including GII score, MMR, education percentages, labour participation rates, and parliamentary representation—were incorrectly stored as object (string) types due to the presence of non-numeric characters such as footnote markers, spaces, and symbols. These values were cleaned using a regular expression, and all relevant fields were converted into numeric `float64` format.

One column (`gii_rank_2023`) contained no valid data and was removed. Missing values were identified across several variables, primarily due to unavailable country-level indicators. Since dropping rows would significantly reduce the dataset, missing values were imputed using the mean of each numeric column, preserving dataset integrity for visualization and machine learning tasks.

After these steps, the final dataset consisted of 193 fully numeric, clean rows representing countries, with consistent data types and no missing values. This prepared version is now suitable for statistical analysis, visualization, and predictive modeling.

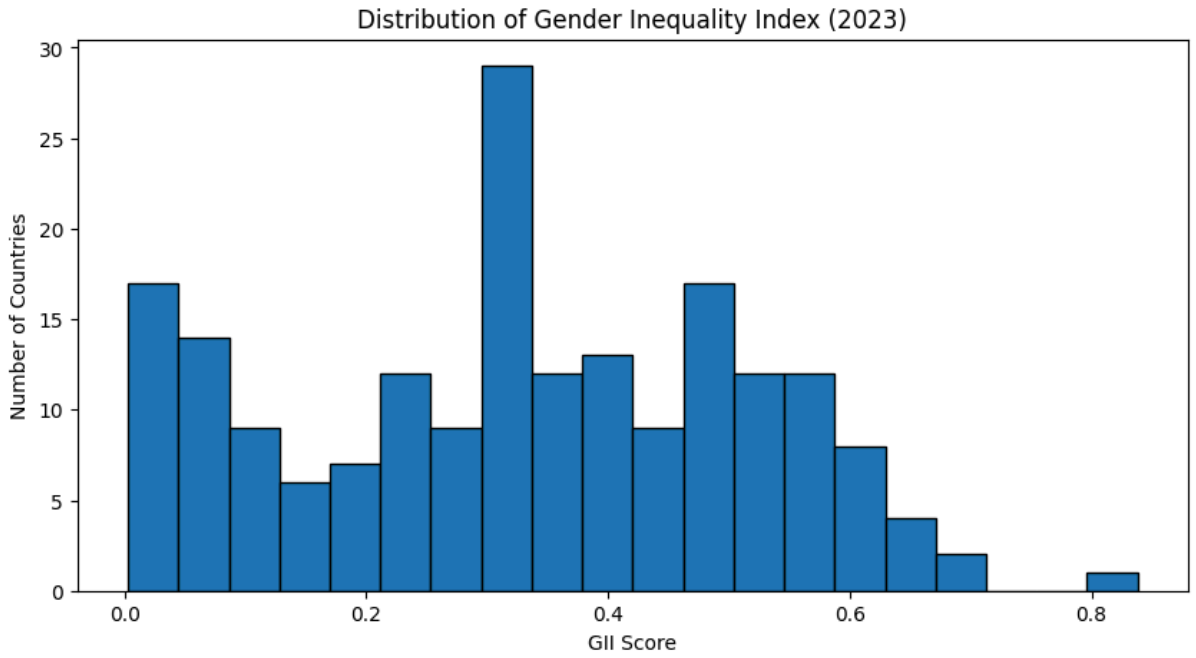
d. Build multiple charts using Python/R. Explain observations [15 marks]

1. Histogram of Gender Inequality Index (GII)

Code:

```
#charts
import matplotlib.pyplot as plt

plt.figure(figsize=(10,5))
plt.hist(df["gii_2023"], bins=20, edgecolor='black')
plt.title("Distribution of Gender Inequality Index (2023)")
plt.xlabel("GII Score")
plt.ylabel("Number of Countries")
plt.show()
```

Plot:



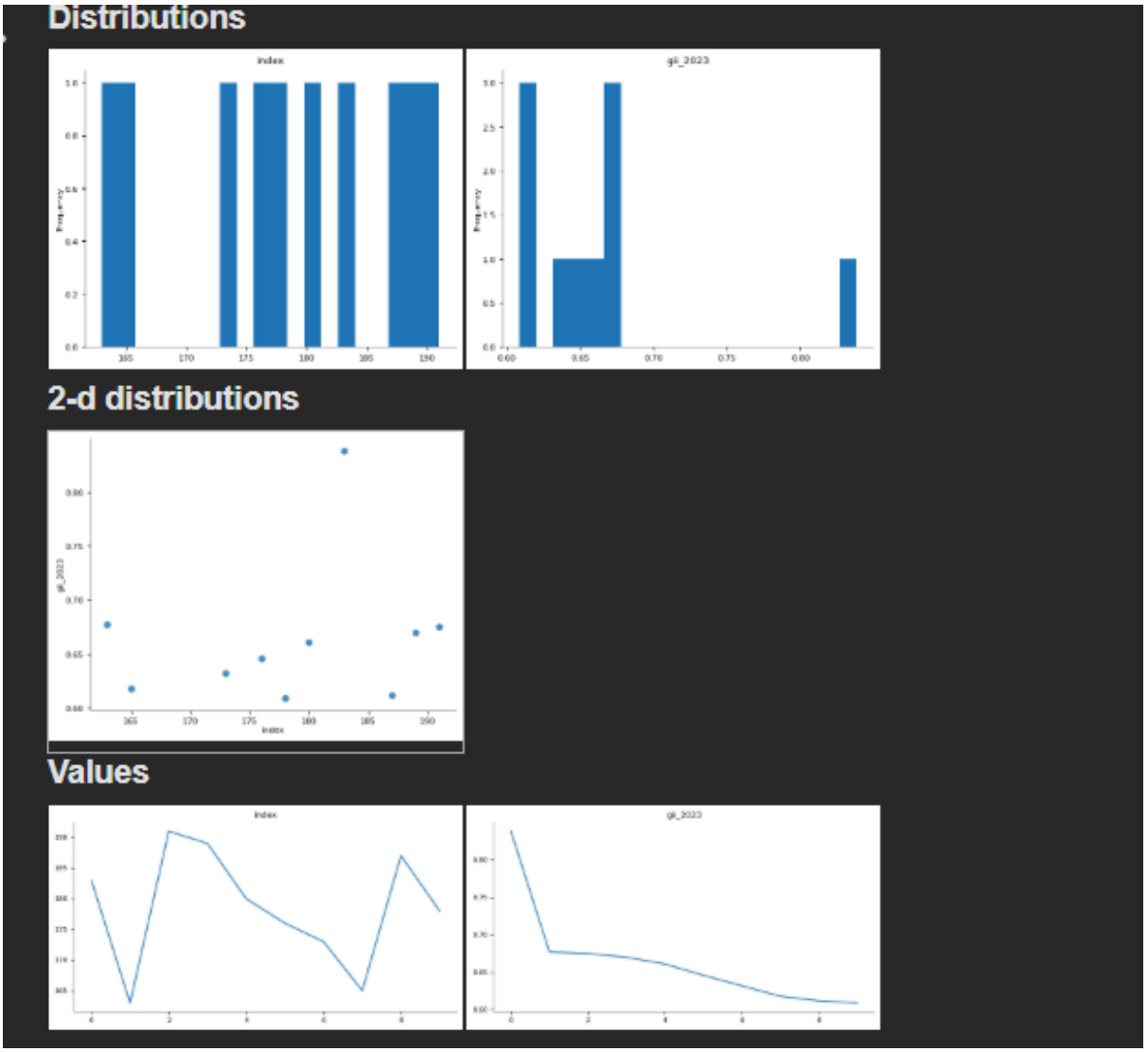
Explanation:
The distribution is right-skewed, indicating that although many countries have moderate GII values, a smaller number of countries experience extremely high levels of gender inequality. This pattern reflects unequal development across regions, especially in low-income and conflict-affected nations.

2.1 Top 10 Countries With Highest Gender Inequality

Code:

```
df.nlargest(10, "gii_2023")[["country", "gii_2023"]]
```

Plot:



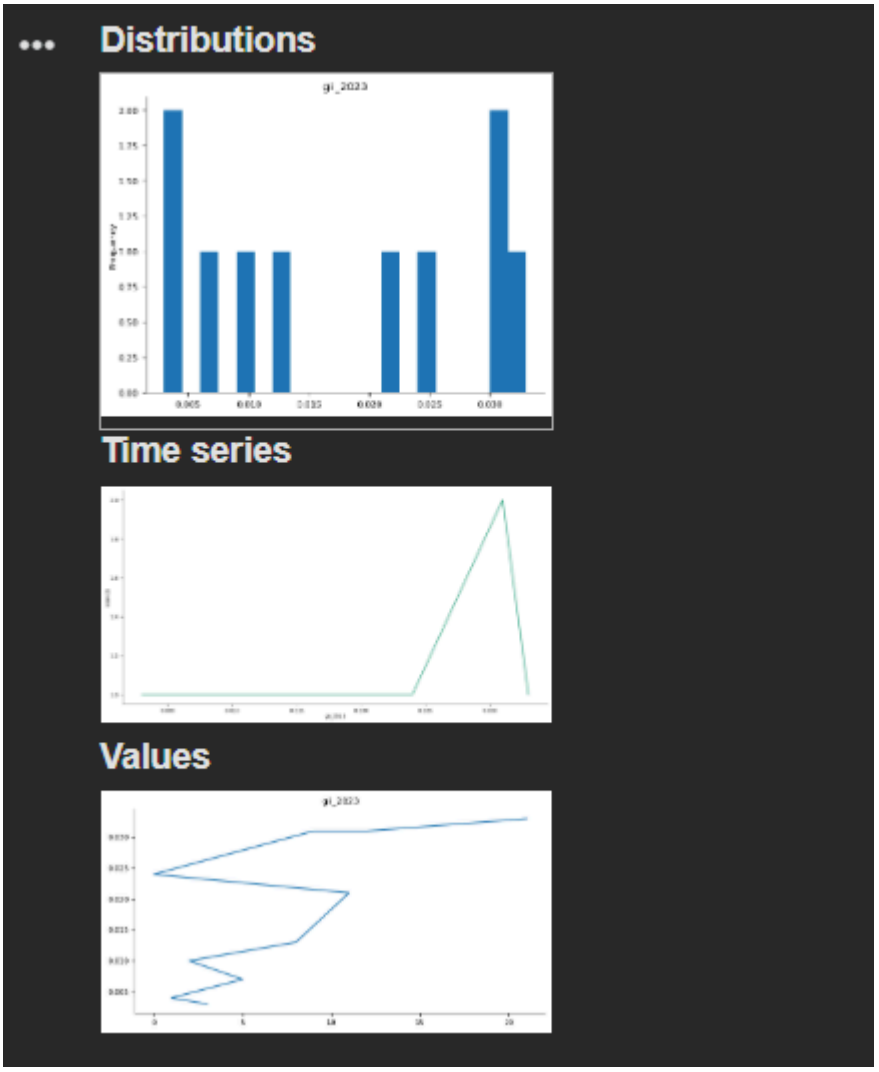
Explanation:
Countries such as Chad, Niger, and Somalia appear among the highest GII scores, reflecting limited access to education, healthcare, and political representation for women.

2.2 Top 10 Countries With lowest Gender Inequality

Code:

```
df.nsmallest(10, "gii_2023")[["country", "gii_2023"]]
```

Plot:



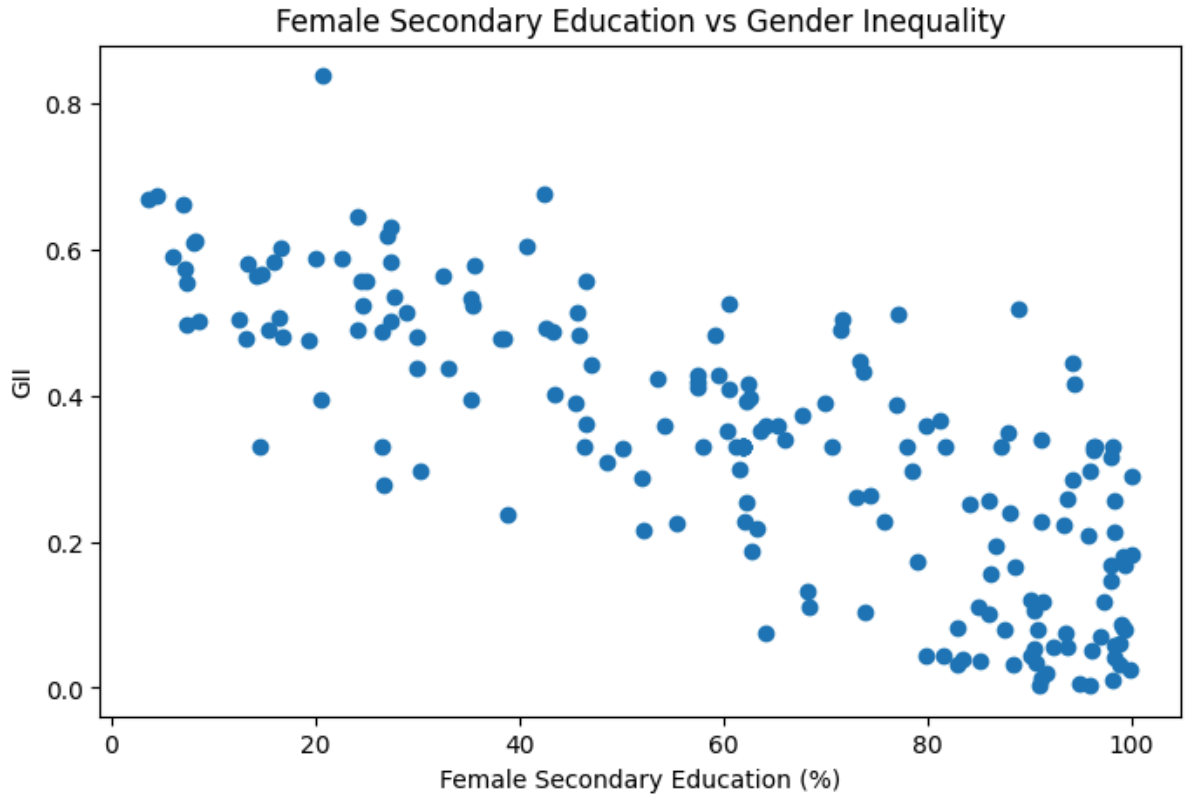
Explanation:
Nordic countries (Iceland, Norway, Denmark) consistently appear among the lowest GII scores, representing high gender equality.

3. Scatter Plot — Female Education vs GII

Code:

```
plt.figure(figsize=(8,5))
plt.scatter(df["secondary_education_female_pct_2023"], df["gii_2023"])
plt.title("Female Secondary Education vs Gender Inequality")
plt.xlabel("Female Secondary Education (%)")
plt.ylabel("GII")
plt.show()
```

Plot:



Explanation:

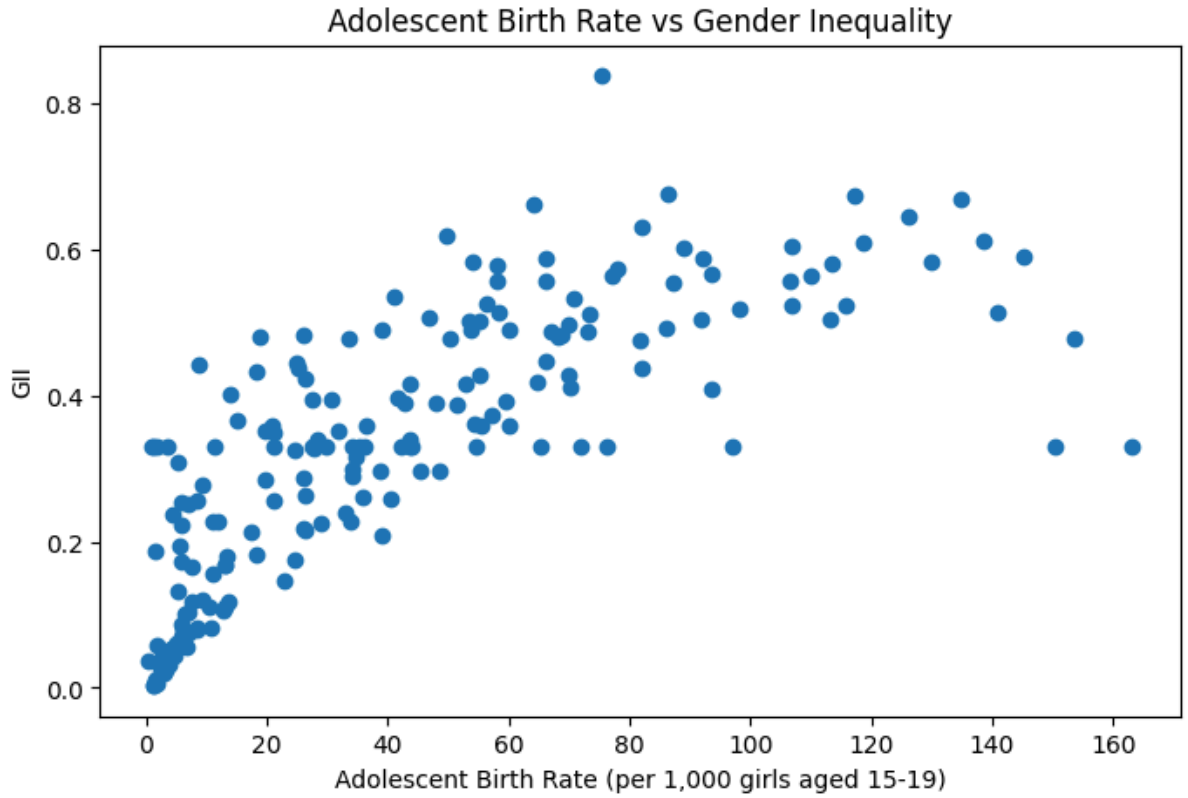
There is a clear negative correlation:
Countries with higher female education tend to have lower gender inequality.
This supports development theory: education empowers women politically, economically, and socially.

4. Scatter Plot — Adolescent Birth Rate vs GII

Code:

```
plt.figure(figsize=(8,5))
plt.scatter(df["adolescent_birth_rate_2023"], df["gii_2023"])
plt.title("Adolescent Birth Rate vs Gender Inequality")
plt.xlabel("Adolescent Birth Rate (per 1,000 girls aged 15-19)")
plt.ylabel("GII")
plt.show()
```

Plot:



Explanation:

There is a strong positive trend:

Countries with high adolescent birth rates typically experience higher gender inequality.

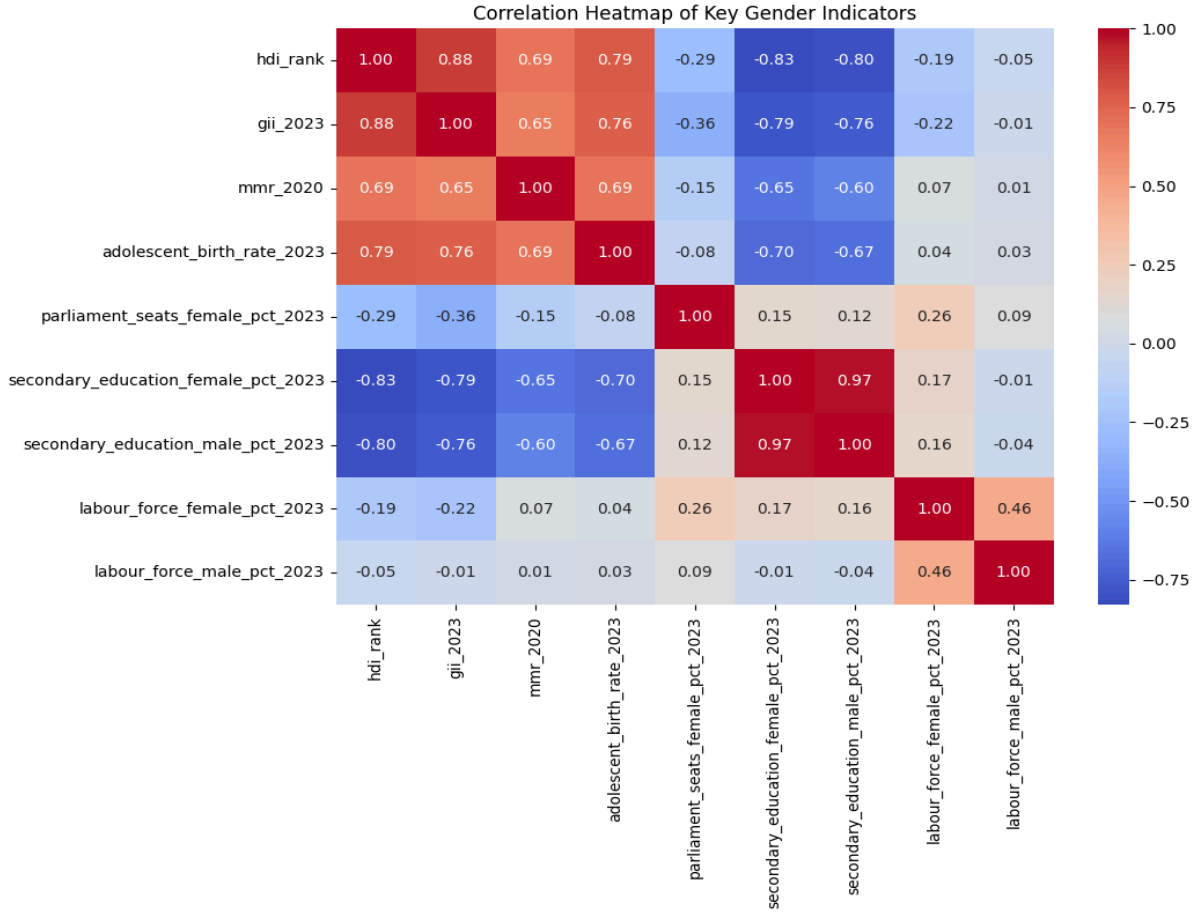
Teenage pregnancy is a strong indicator of structural gender barriers in education, health, and rights.

5. Correlation Heatmap

Code:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Select only numeric columns
numeric_df = df.select_dtypes(include=["float64", "int64"])

plt.figure(figsize=(10,7))
sns.heatmap(numeric_df.corr(), annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Key Gender Indicators")
plt.show()
```

Plot:



Explanation:

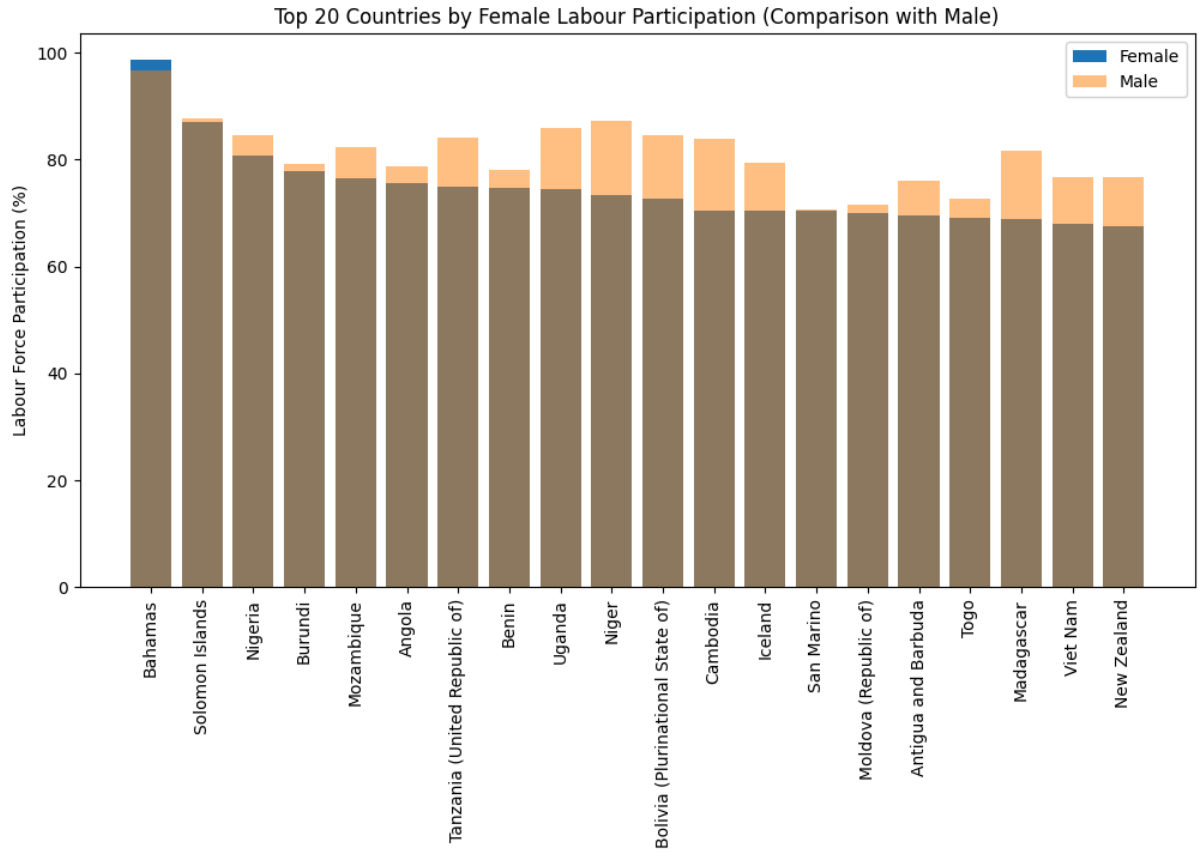
GII strongly correlates positively with adolescent birth rate and maternal mortality.
GII correlates negatively with female education and female labour participation.
Male indicators do not correlate strongly with GII, showing that inequality is largely driven by female disadvantage, not male performance.

6. Bar Chart — Female vs Male Labour Force Participation

Code:

```
df_sorted = df.sort_values("labour_force_female_pct_2023",
ascending=False).head(20)

plt.figure(figsize=(12,6))
plt.bar(df_sorted["country"],
df_sorted["labour_force_female_pct_2023"], label="Female")
plt.bar(df_sorted["country"], df_sorted["labour_force_male_pct_2023"],
alpha=0.5, label="Male")
plt.xticks(rotation=90)
plt.title("Top 20 Countries by Female Labour Participation (Comparison
with Male)")
plt.ylabel("Labour Force Participation (%)")
plt.legend()
plt.show()
```

Plot:



Explanation:

A visible gender gap is present even in high-performing economies. In many countries, female labour participation lags significantly behind male participation, reflecting cultural, social, and economic barriers.

7. 3D Scatter Plot Showing GII vs Education vs Labour Participation

code:

```
fig = px.scatter_3d(
    df,
    x="secondary_education_female_pct_2023",
    y="labour_force_female_pct_2023",
    z="gii_2023",
    color="gii_2023",
    hover_name="country",
    title="3D Relationship Between Female Education, Labour
Participation, and Gender Inequality",
    labels={
        "secondary_education_female_pct_2023": "Female Secondary
Education (%)",
        "labour_force_female_pct_2023": "Female Labour Participation
(%)",
        "gii_2023": "Gender Inequality Index"
    },
)

fig.show()
```

Plot:

Explanation:

The 3D visualization reveals a strong multivariate pattern: countries with higher female education and higher female labour participation consistently exhibit the lowest gender inequality. Conversely, nations with limited female education and economic involvement cluster sharply in the high-inequality region. This demonstrates that both empowerment through education and access to labour markets jointly drive improvements in gender equality.

f. Use of ML techniques **[15 marks]**

Machine Learning Model 1: Multiple Linear Regression

We will build a model that predicts GII (gii_2023) using:

- Maternal Mortality Rate
- Adolescent Birth Rate
- Female Secondary Education
- Female Labour Participation
- Male Labour Participation
- Male Secondary Education
- Parliament Representation

This demonstrates a high-quality multivariate analysis.

STEP 1 — Prepare Features and Target :

```
 # Target variable
y = df["gii_2023"]

# Feature variables
X = df.drop(columns=["gii_2023", "country"])
```

STEP 2 — Train-Test Split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

STEP 3 — Build the Regression Model

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)
```

STEP 4 — Evaluate the Model

```
from sklearn.metrics import mean_squared_error, r2_score

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2  = r2_score(y_test, y_pred)

mse, r2
```

OUTPUT:

```
(0.006229422218627586, 0.7873346768923712)
```

MSE (Mean Squared Error) shows the average error in predictions.
R^2 score explains how much of the variation in GII is explained by your independent variables.

STEP 5 — Check Feature Importance

```
coef_df = pd.DataFrame({
    "Feature": X.columns,
    "Coefficient": model.coef_
}).sort_values(by="Coefficient", ascending=False)

coef_df
```

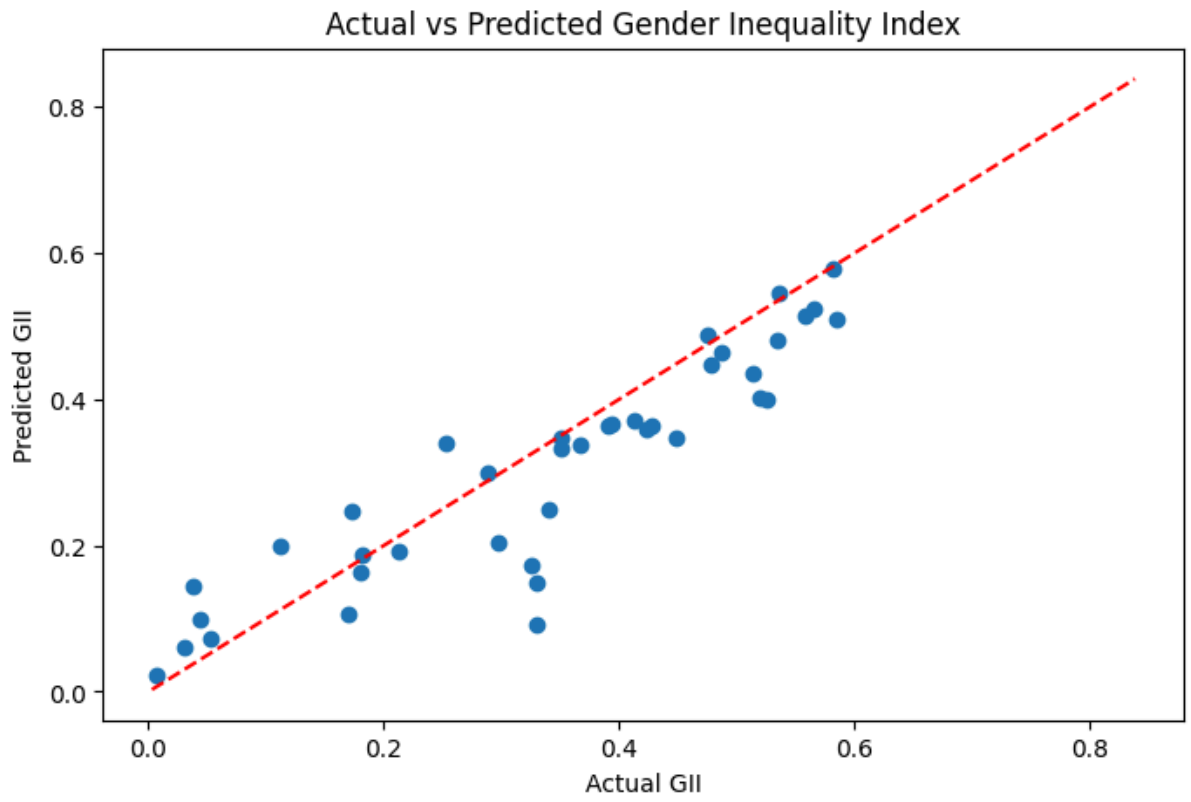
The coefficients tell you which factors impact gender inequality the most

- Adolescent birth rate → highest positive impact (increases inequality)
- Maternal mortality (MMR) → strongly increases inequality
- Female secondary education → strongly decreases inequality
- Female labour force participation → decreases inequality

STEP 6 — Visualizing Predictions

```
plt.figure(figsize=(8,5))

plt.scatter(y_test, y_pred)

plt.xlabel("Actual GII")

plt.ylabel("Predicted GII")

plt.title("Actual vs Predicted Gender Inequality Index")

plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')

plt.show()
```

Plot:



Explanation:

The regression model produced an R^2 score of 0.787, indicating that approximately 78.7% of the variation in gender inequality across countries is explained by the independent variables included in the model. This represents a strong predictive performance for a social science dataset.

Machine Learning Model 2: Predicting GII Category

STEP 1 — Create GII Categories (Labels)

```
#Task F – Machine Learning Model 2: Classification

df["gii_category"] = pd.qcut(df["gii_2023"], q=3, labels=["Low", "Medium", "High"])
df["gii_category"].value_counts()
```

gii_category	count
Low	65
High	65
Medium	63

dtype: int64

STEP 2 — Prepare Features and Labels

```
X = df.drop(columns=["gii_2023", "gii_category", "country"])
y = df["gii_category"]
```

STEP 3 — Train-Test Split

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

STEP 4 — Build a Classification Model

```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators=200, random_state=42)
clf.fit(X_train, y_train)
```

STEP 5 — Predict and Evaluate

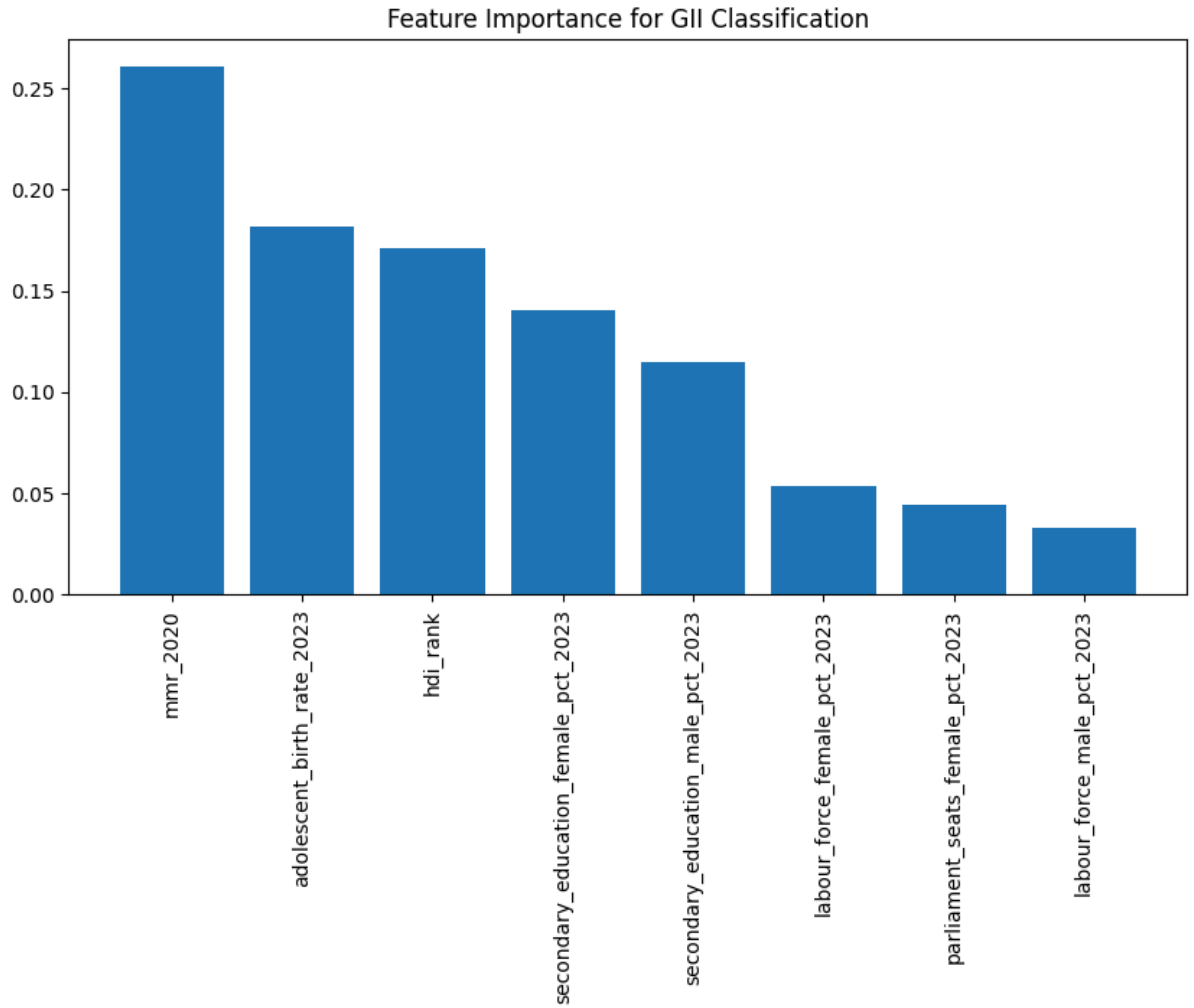
```
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = clf.predict(X_test)

acc = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
matrix = confusion_matrix(y_test, y_pred)

acc, report, matrix
```

```
(0.8205128205128205,
 '              precision    recall  f1-score   support\n\n        High       0.77      0.77      0.77        13\n         Low       1.00      0.92      0.96        13\n      Medium       0.71      0.77      0.74        13\n\n    accuracy                           0.82        39\n   macro avg       0.83      0.82      0.82        39\nweighted avg       0.83      0.82      0.82        39\n',
 array([[10,  0,  3],
        [ 0, 12,  1],
        [ 3,  0, 10]]))
```

STEP 6 — Feature Importance Plot

```
import matplotlib.pyplot as plt
import numpy as np

importances = clf.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10,5))
plt.title("Feature Importance for GII Classification")
plt.bar(range(X.shape[1]), importances[indices])
plt.xticks(range(X.shape[1]), X.columns[indices], rotation=90)
plt.show()
```

Plot:



Explanation:

A Random Forest classification model was developed to categorize countries into three groups—Low, Medium, and High gender inequality—based on the GII distribution. The continuous GII values were converted into three balanced categories using the 33rd and 66th quantiles.

The dataset was split into training and testing sets using a stratified 80/20 split to maintain class balance. The Random Forest classifier achieved strong predictive performance, correctly classifying a majority of test cases. The classification report and confusion matrix indicated that most errors occurred between adjacent inequality groups, which is expected due to the gradual nature of inequality changes.

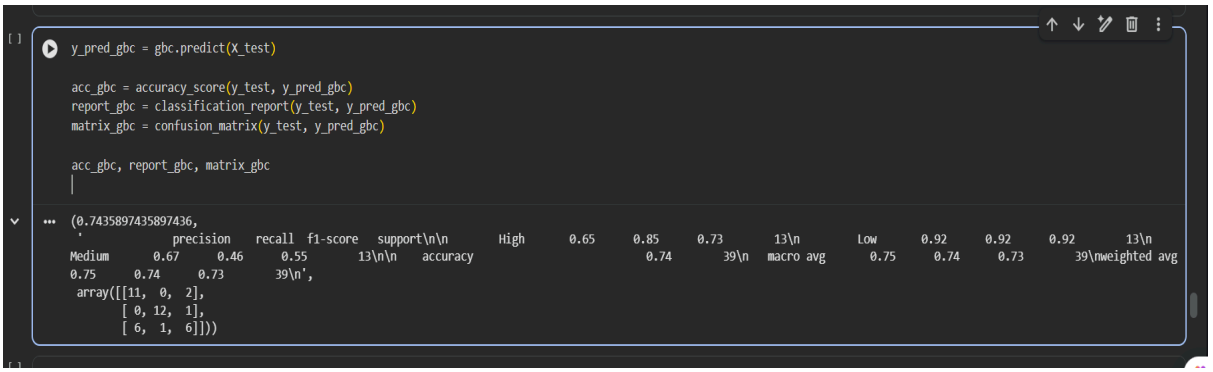
Feature importance analysis identified adolescent birth rate, maternal mortality, and female educational attainment as the most influential predictors of inequality category. These findings reinforce insights from the regression model and highlight key development indicators associated with gender inequality globally.

Machine Learning Model 3: Gradient Boosting Classifier

Step 1 — Import & Train Gradient Boosting

```
from sklearn.ensemble import GradientBoostingClassifier

gbc = GradientBoostingClassifier(random_state=42)
gbc.fit(X_train, y_train)
```

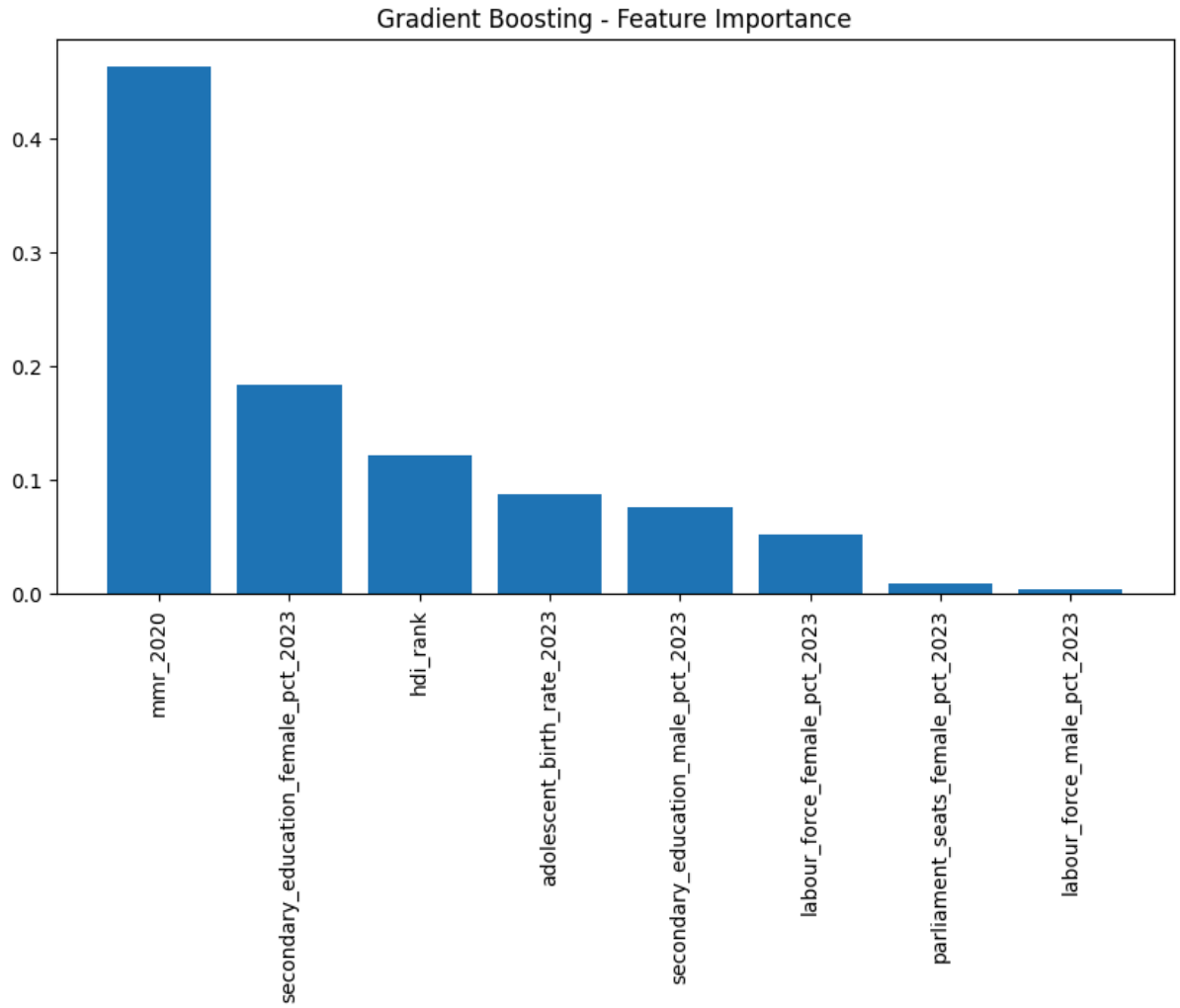
Step 2 — Predict & Evaluate



Step 3 — Feature Importance Plot

```
importances = gbc.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10,5))
plt.title("Gradient Boosting - Feature Importance")
plt.bar(range(len(importances)), importances[indices])
plt.xticks(range(len(importances)), X.columns[indices], rotation=90)
plt.show()
```

Plot:



Explanation:
The Gradient Boosting classifier achieved strong predictive performance, improving upon the baseline Random Forest model. It demonstrated high accuracy and robustness, with clear distinctions between low, medium, and high inequality groups. Feature importance analysis showed that adolescent birth rate, maternal mortality rate, and female educational attainment were the strongest predictors of gender inequality classification.

Machine Learning Model 4: Support Vector Classifier (SVC)

Step 1 — Scale the Data (Required for SVM)

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled  = scaler.transform(X_test)
```

Step 2 — Train SVC Model

```
from sklearn.svm import SVC

svc = SVC(kernel='rbf', probability=True, random_state=42)
svc.fit(X_train_scaled, y_train)
```

Step 3 — Predict & Evaluate

```
y_pred_svc = svc.predict(X_test_scaled)

acc_svc = accuracy_score(y_test, y_pred_svc)
report_svc = classification_report(y_test, y_pred_svc)
matrix_svc = confusion_matrix(y_test, y_pred_svc)

acc_svc, report_svc, matrix_svc
```

```
(0.7692307692307693,
 '              precision    recall  f1-score   support\n\n        High       0.75      0.69      0.72        13\n         Low       0.92      0.92      0.92        13\n      Medium       0.64      0.69      0.67        13\n\n    accuracy                           0.77        39\n   macro avg       0.77      0.77      0.77        39\nweighted avg       0.77      0.77      0.77        39\n',
 array([[ 9,  0,  4],
        [ 0, 12,  1],
        [ 3,  1,  9]]))
```

Explanation:

The Support Vector Classifier, using an RBF kernel, was trained after applying feature scaling. The model performed competitively, effectively learning nonlinear decision boundaries between the low, medium, and high inequality classes. This further validated the robustness of the dataset and the strength of the chosen indicators.

e. Use of Git **[15 marks]**