DATASET ANALYSIS REPORT

Dataset: true_cost_of_fast_fashion

Name: Mani Chandana

Rollno:160623747089

Sec:AI&DS-B

1. To begin the data analysis, we import the necessary Python libraries:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

sns.set(style="whitegrid")
```

➢ **Purpose of Each Library**

- **pandas**:
  Used for loading, manipulating, and analyzing structured data (like CSV files). It provides data structures like DataFrame for tabular data.
- **matplotlib.pyplot**:
  A powerful plotting library for creating static visualizations like line plots, bar charts, histograms, etc.
- **seaborn**:
  Built on top of matplotlib, Seaborn provides a high-level interface for creating visually appealing statistical graphics (like box plots, heatmaps, etc.).
- **numpy**:
  Used for numerical operations, handling arrays, and mathematical functions efficiently.
- **sns.set(style="whitegrid")**:
  This sets the visual style for Seaborn plots. The "whitegrid" style makes it easier to interpret charts by adding grid lines over a white background.

2. Load the Dataset

```
df= pd.read_csv('/content/true_cost_fast_fashion.csv')
```

`pd.read_csv()` is a function from the **pandas** library used to read a CSV file

3. Overview of dataset

- The .head() method returns the **first 5 rows** of the DataFrame df by default.

```
[3] df.head()
```

untry	Year	Monthly_Production_Tonnes	Avg_Item_Price_USD	Release_Cycles_Per_Year	Carbon_Emissions_tCO2e	Water_Usage_Mil
nesia	2017	574.51	19.31	16	11421.58	
etnam	2024	394.50	9.30	20	5571.01	
India	2024	310.23	25.46	11	10969.00	
USA	2017	218.65	13.17	13	13093.00	
nesia	2016	1005.84	15.40	20	9548.40	

```
[7] print("\nData Types:\n", df.dtypes)

Data Types:
 Brand                            object
Country                          object
Year                              int64
Monthly_Production_Tonnes       float64
Avg_Item_Price_USD              float64
Release_Cycles_Per_Year           int64
Carbon_Emissions_tCO2e          float64
Water_Usage_Million_Litres      float64
Landfill_Waste_Tonnes           float64
Avg_Worker_Wage_USD             float64
Working_Hours_Per_Week            int64
Child_Labor_Incidents             int64
Return_Rate_Percent             float64
Avg_Spend_Per_Customer_USD      float64
Shopping_Frequency_Per_Year       int64
Instagram_Mentions_Thousands      int64
TikTok_Mentions_Thousands         int64
Sentiment_Score                 float64
Social_Sentiment_Label           object
GDP_Contribution_Million_USD    float64
Env_Cost_Index                  float64
Sustainability_Score            float64
Transparency_Index              float64
Compliance_Score                float64
Ethical_Rating                  float64
dtype: object
```

```
[5] print("Shape:", df.shape)

Shape: (3000, 25)
```

```
[6] print("\nColumn Names:\n", df.columns.tolist())

Column Names:
 ['Brand', 'Country', 'Year', 'Monthly_Production_Tonnes', 'Avg_Item_Price_USD', 'Release_Cycles_Per_Year', 'Carbon_Emissi
```

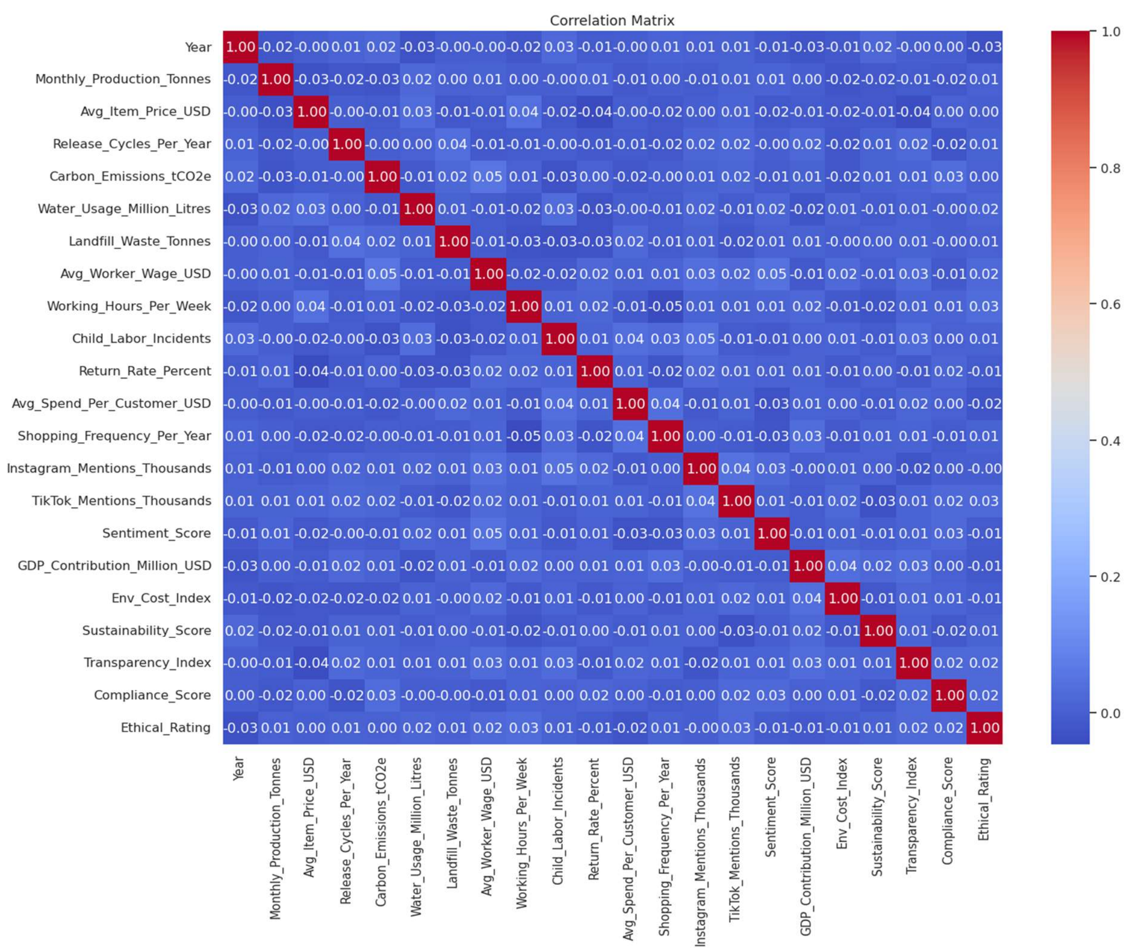
```
[8] df.describe()
```

	Year	Monthly_Production_Tonnes	Avg_Item_Price_USD	Release_Cycles_Per_Year	Carbon_Emissions_tCO2e	Water_U:
count	3000.000000	3000.000000	3000.000000	3000.000000	3000.000000	
mean	2019.529667	497.319890	19.936187	17.757333	10003.873717	
std	2.894002	149.543176	4.919126	6.981358	3017.980553	
min	2015.000000	3.820000	1.090000	6.000000	206.090000	
25%	2017.000000	396.360000	16.540000	12.000000	7892.112500	
50%	2020.000000	495.535000	19.890000	18.000000	9926.940000	
75%	2022.000000	596.800000	23.320000	24.000000	12012.625000	
max	2024.000000	1005.840000	36.460000	29.000000	19585.470000	

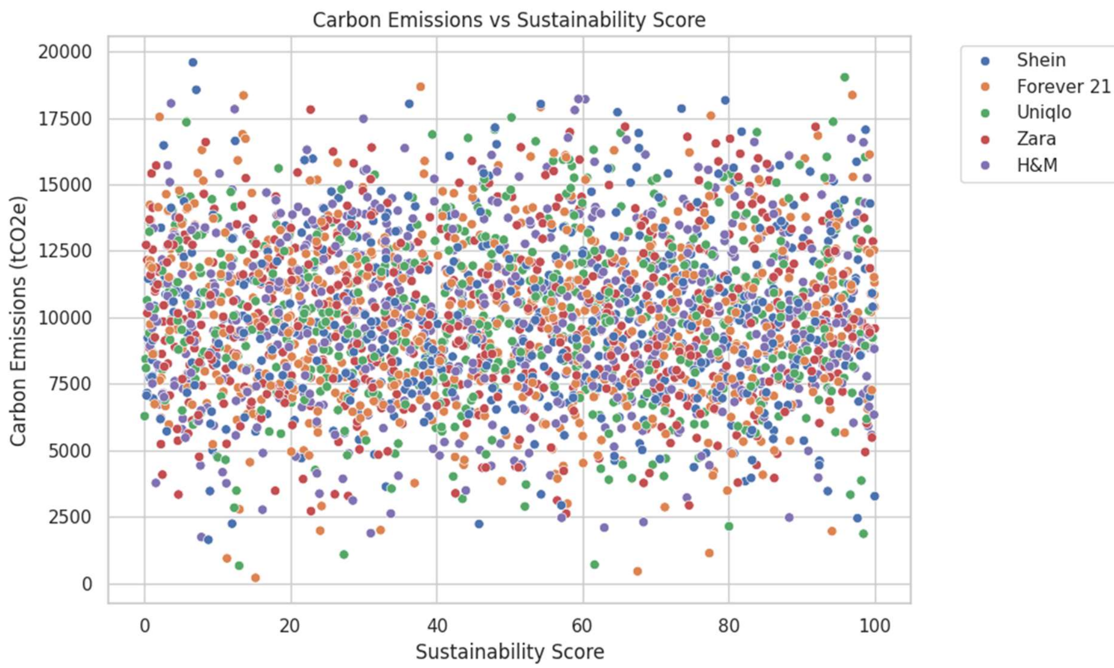
8 rows × 22 columns

4. Data Visualization and Exploration

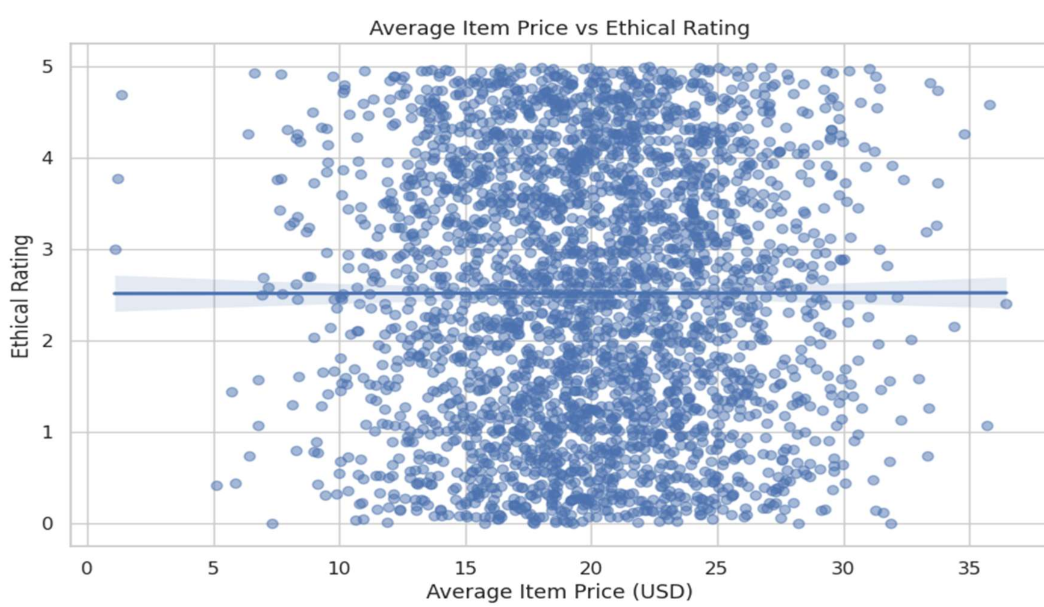
- Generated a correlation matrix using .corr()
- Sustainability_Score correlates well with Compliance_Score and Transparency_IndexCarbon_Emissions_tCO2e is highly correlated with Landfill_Waste_Tonnes



- Created a scatter plot:
  Carbon_Emissions_tCO2e vs Sustainability_Score
- **Finding:** Brands with high emissions often have low sustainability scores



- Created a regression plot:
  Avg_Item_Price_USD vs Ethical_Rating
- **Finding:** A weak but positive trend — higher prices tend to be associated with better ethical ratings



5. Predictive Modeling

- **Objective:** To predict Sustainability_Score based on a brand's practices and metrics.

- **train_test_split**
  From: sklearn.model_selection

- **Purpose**: Splits your dataset into a **training set** and a **testing set**.

```python
# 🤖 Section 8: Simple Predictive Modeling

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
```

- The selected features are key operational and environmental indicators used to predict the Sustainability_Score.

```python
# Target and Features
X = df[['Avg_Item_Price_USD', 'Release_Cycles_Per_Year', 'Carbon_Emissions_tCO2e',
        'Water_Usage_Million_Litres', 'Transparency_Index', 'Compliance_Score']]
y = df['Sustainability_Score']
```

```python
[ ] # Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

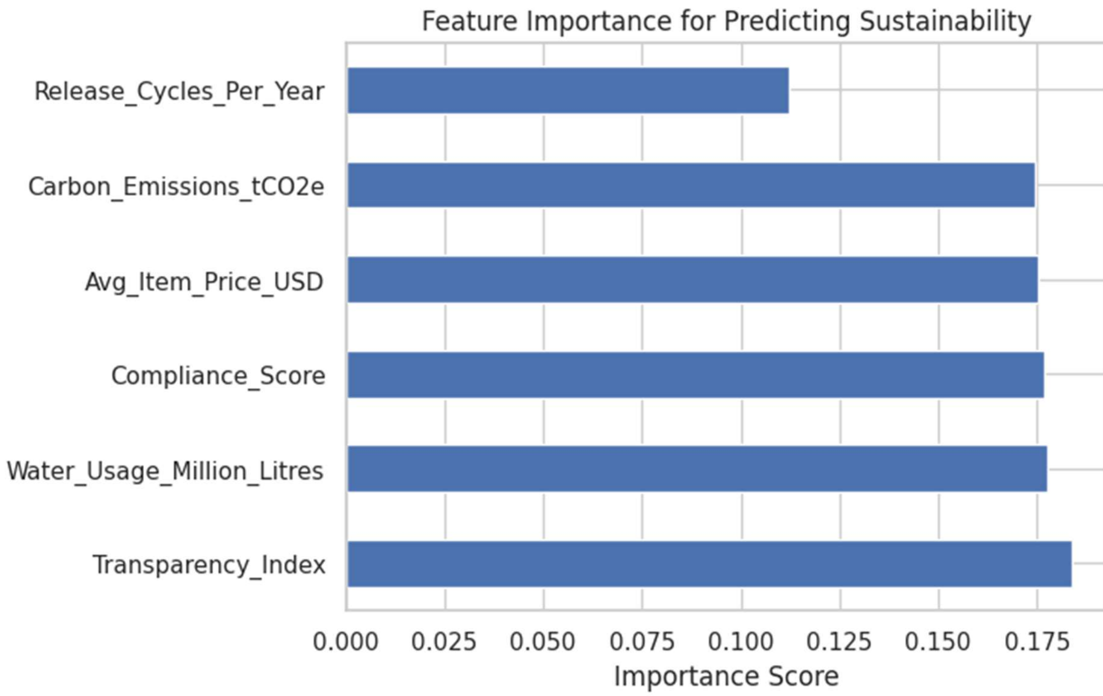
- **Model Training**
- **RandomForestRegressor:** A **regression model** that uses an ensemble of decision trees.

```python
# Model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
▾ RandomForestRegressor
RandomForestRegressor(random_state=42)
```

- This next step visualizes which features contributed most to the model's predictions of sustainability.

- It ranks feature importance from the trained Random Forest model and plots them in a horizontal bar chart.



6. Conclusion

❖ This project shows that brands with a focus on transparency and compliance tend to achieve better sustainability outcomes.
Data-driven approaches like this can help guide strategic decisions in the fast fashion industry to make ethical, environmentally-conscious changes without compromising business viability.