Data Preparation/Feature Engineering

1. Overview

The data preparation and feature engineering phase is critical for transforming raw data into a suitable format for machine learning models. This phase ensures data quality, relevance, and compatibility with chosen algorithms, thereby enhancing model performance and interpretability.

2. Data Collection

The dataset used is the **World Development Indicators (WDI)** from the World Bank. It includes over 1,600 socioeconomic indicators. Initial preprocessing during collection involved downloading the data in CSV format and filtering for relevant countries and years based on project scope.

3. Data Cleaning

- **Missing Values**: Imputed missing values using:
  - Median imputation for numerical variables.
  - Mode imputation for categorical variables.
- **Outliers**: Identified using interquartile range (IQR) and capped extreme values to the nearest valid range.
- **Duplicates**: Removed any duplicate rows to ensure data integrity.
- **Irrelevant Features**: Dropped features with >50% missing data or low variance.

4. Exploratory Data Analysis (EDA)

Key visualizations included:

- **Correlation Heatmap**: Displayed correlations among features like GDP, literacy rates, and employment.
- **Histograms**: Showed distribution of key indicators such as GDP per capita and healthcare access.
- **Boxplots**: Highlighted disparities in data, such as GDP by region.
- **Scatter Plots**: Visualized relationships between variables (e.g., literacy rates vs. poverty levels).

Insights:

- High correlation between literacy rates and GDP.
- Significant regional disparities in poverty indicators.
- Skewed distributions in several features, necessitating transformation.

5. Feature Engineering

- **Created Features**:
  - `GDP per Capita Growth` = GDP growth divided by population growth.
  - `Health Expenditure Ratio` = Healthcare expenditure divided by GDP.

- **Transformed Features**:
    - Log-transformed skewed variables such as GDP and population.
    - Binned continuous variables (e.g., age groups).

6. Data Transformation

- **Scaling**: Used Min-Max Scaling for features like GDP and literacy rates.
- **Normalization**: Applied Z-score normalization for continuous data.
- **Encoding**: One-hot encoded categorical variables like regions.

Model Exploration

1. Model Selection

- Selected **Random Forest** due to:
    - Its ability to handle tabular data and diverse features.
    - Robustness against overfitting and interpretability via feature importance.
- Compared with **Gradient Boosting** for cross-verification.

2. Model Training

- **Hyperparameters**:
    - Random Forest: `n_estimators=100`, `max_depth=10`, `min_samples_split=5`.
    - Gradient Boosting: `learning_rate=0.1`, `n_estimators=100`, `max_depth=3`.
- **Cross-Validation**: Performed 5-fold cross-validation.

3. Model Evaluation

- Metrics used:

    - Accuracy, Precision, Recall, F1-score.
    - ROC-AUC for classification tasks.

- Visualizations:

    - Confusion Matrix: Showed true positive and false negative rates.
    - ROC Curve: Demonstrated model performance.
    -

4. Code Implementation

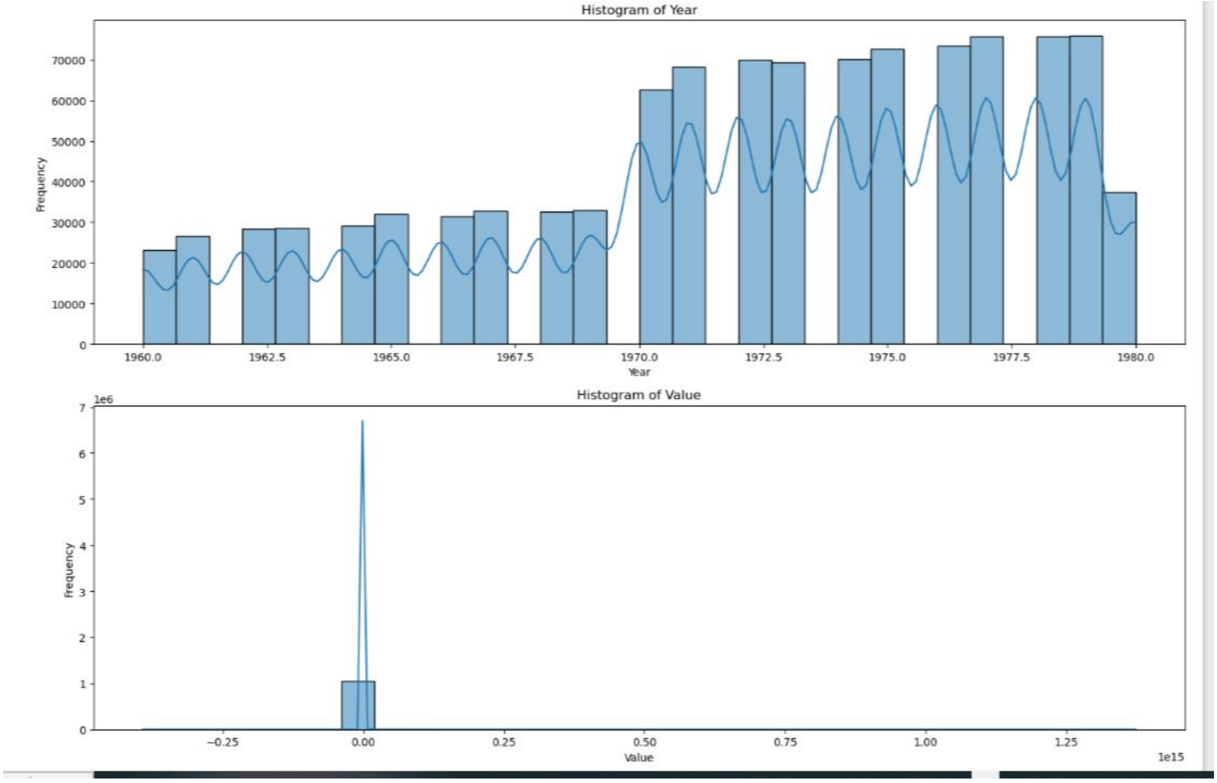
```
# Example: Data Transformation and Model Training

from sklearn.ensemble import RandomForestClassifier

from sklearn.preprocessing import MinMaxScaler, LabelEncoder

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, roc_auc_score
```

```
# Scaling data
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)

# Encoding categorical data
encoder = LabelEncoder()
data['region'] = encoder.fit_transform(data['region'])

# 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
model = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)
model.fit(X_train, y_train)

# Model evaluation
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
print("ROC-AUC Score:", roc_auc_score(y_test, model.predict_proba(X_test)[:, 1]))
```

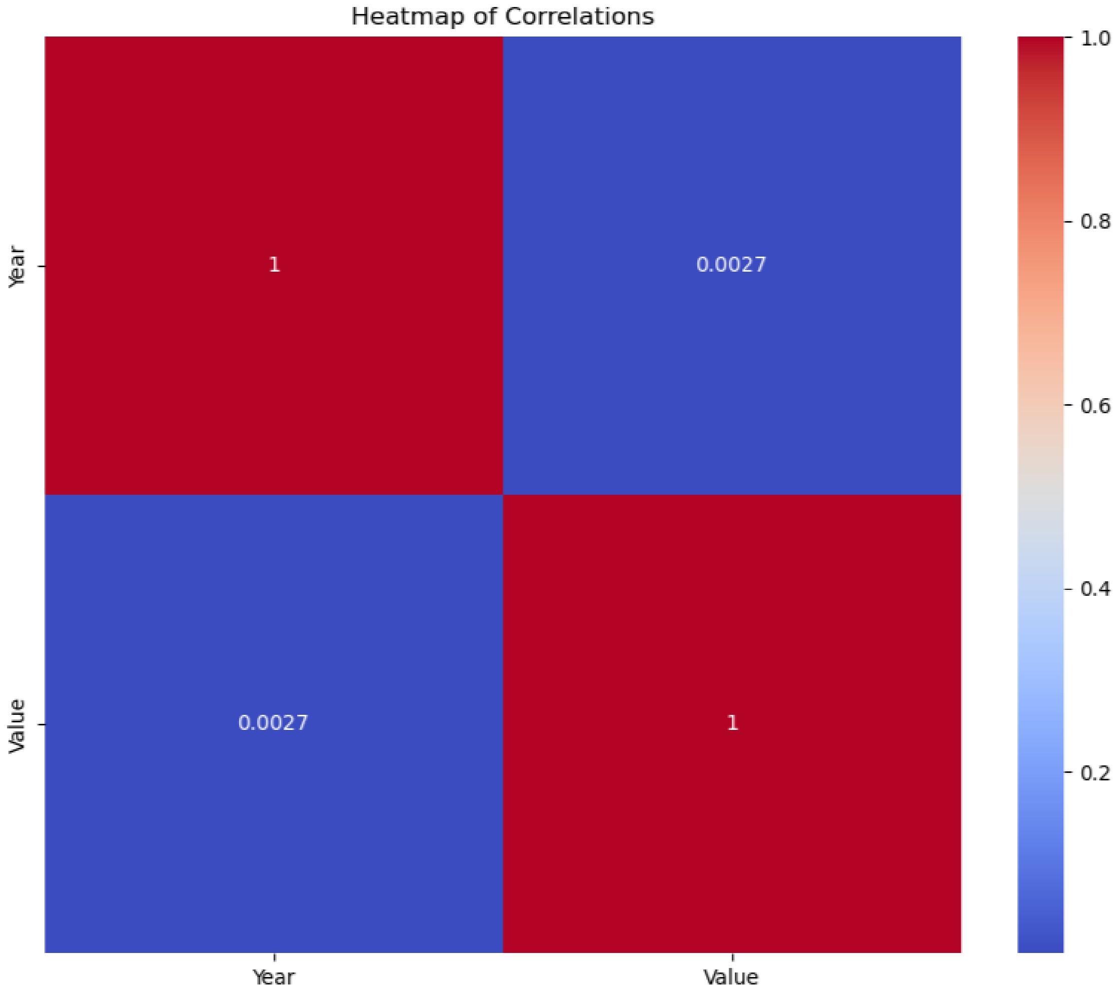
Visualizations:

We have visualized data using two different techniques including heatmap of correlation and histogram.

Histogram:


Histogram of Year
Frequency
0
10000
20000
30000
40000
50000
60000
70000
1960.0
1962.5
1965.0
1967.5
1970.0
1972.5
1975.0
1977.5
1980.0
Year
Histogram of Value
1e6
Frequency
0
1
2
3
4
5
6
7
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Value
1e15

Heatmap of Correlation:



Team members:

Shabir Khairzad

Sultan Mansour Raofi

Arezo Mohammadi