IMPROVING DATA ACCURACY IN CRM USING AI

Phase 2 | DATA EXPLORATION & SOLUTION ARCHITECTURE

Submitted by:

Aditya Mahesh Patil (2VD21CS001)

Amarnath Mahesh Patil (2VD21CS006)

Hrishikesh M Agnihotri (2VD21CS018)

Nivedita G Nayak(2VD21CS033)

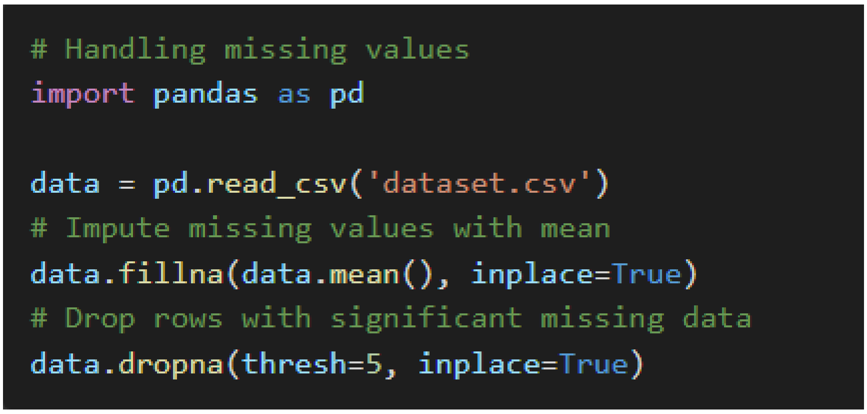
1. Overview of Data Visualization and Analysis

Data visualization and analysis are critical steps in understanding and interpreting datasets. They help uncover patterns, trends, and insights that can guide decision-making processes. This report outlines the steps taken to clean, prepare, and visualize data, as well as the methods used for analysis and feature engineering.

2. Data Cleaning and Preparation

2.1 Handling Missing Values

Missing values in the dataset were identified and addressed using the following techniques:



```python
# Handling missing values
import pandas as pd

data = pd.read_csv('dataset.csv')
# Impute missing values with mean
data.fillna(data.mean(), inplace=True)
# Drop rows with significant missing data
data.dropna(thresh=5, inplace=True)
```

*dataset.csv renamed from GOI.csv

- **Imputation**: Replacing missing values with the mean, median, or mode.
- **Deletion**: Removing rows or columns with a high proportion of missing data.

2.2 Managing Outliers

Outliers were detected using statistical methods such as the interquartile range (IQR) and visualizations like box plots. Detected outliers were either:

```
# Detecting outliers using IQR
Q1 = data.quantile(0.25)
Q3 = data.quantile(0.75)
IQR = Q3 - Q1
# Removing outliers
cleaned_data = data[~((data < (Q1 - 1.5 * IQR)) | (data > (Q3 + 1.5 *
IQR))).any(axis=1)]
```

2.3 Resolving Duplicates and Inconsistencies

Duplicate entries were identified and removed. Data inconsistencies, such as varying formats or units, were standardized to ensure uniformity.

```
# Removing duplicates
data.drop_duplicates(inplace=True)
# Standardizing column formats
data.columns = data.columns.str.lower().str.replace(' ', '_')
```

3. Data Visualization

3.1 Tools for Visualization

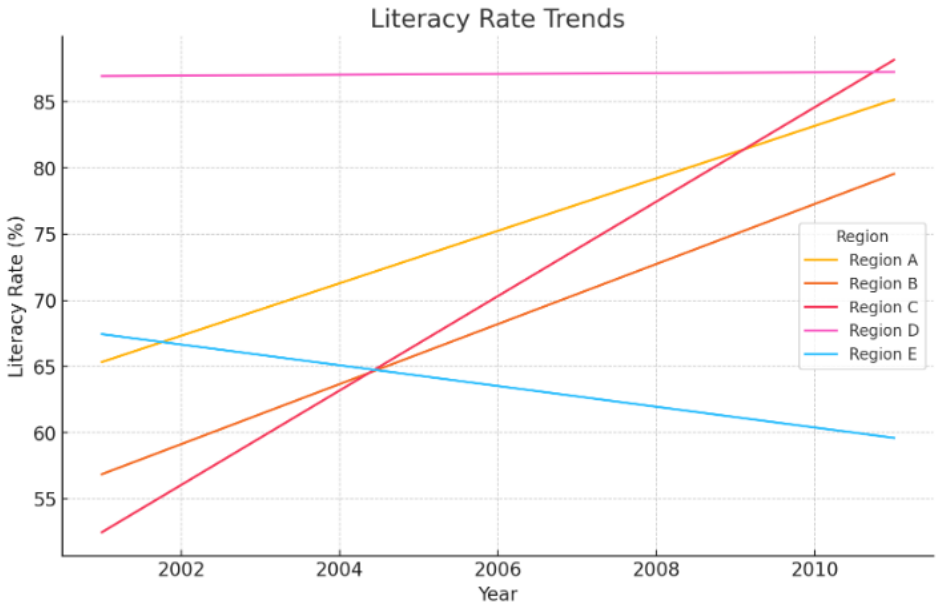
The following tools were utilized for data visualization:

- **Matplotlib** and **Seaborn**: For static visualizations.
- **Plotly**: For interactive charts.
- **Tableau**: For comprehensive dashboards.

3.2 Key Visualizations and Insights

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

# Visualizing literacy trends
plt.figure(figsize=(10, 6))
sns.lineplot(data=cleaned_data, x='year', y='literacy_rate', hue='region')
plt.title('Literacy Rate Trends')
plt.xlabel('Year')
plt.ylabel('Literacy Rate (%)')
plt.show()
```



- **Trend Analysis**: Line charts were used to show literacy rate trends over time.
- **Comparison**: Bar charts highlighted differences between rural and urban literacy rates.
- **Geospatial Analysis**: Maps were used to visualize regional disparities in literacy rates.

4. Model Research and Selection Rationale

4.1 Research into Techniques

Various AI and statistical techniques were researched to improve data accuracy and insights:

```
from sklearn.linear_model import LinearRegression

# Regression for missing value prediction
model = LinearRegression()
model.fit(X_train, y_train)
predicted_values = model.predict(X_test)
```

- **Regression Models**: To predict missing values.
- **Clustering Algorithms**: To group similar data points.

- **Anomaly Detection**: To identify inconsistencies in the dataset.

5. Data Transformation and Feature Engineering

5.1 Feature Scaling

Numerical features were scaled using:

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Min-Max Scaling
scaler = MinMaxScaler()
data_scaled = scaler.fit_transform(data)

# Standard Scaling
scaler = StandardScaler()
data_standardized = scaler.fit_transform(data)
```

5.2 Encoding Categorical Variables

Categorical variables were encoded using:

```
# One-Hot Encoding
encoded_data = pd.get_dummies(data, columns=['category'], drop_first=True)

# Label Encoding
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder()
data['category_encoded'] = encoder.fit_transform(data['category'])
```

5.3 Dimensionality Reduction

Techniques like **Principal Component Analysis (PCA)** were applied to reduce the dimensionality of the dataset while retaining significant variance.

```
from sklearn.decomposition import PCA

# Applying PCA
pca = PCA(n_components=2)
data_reduced = pca.fit_transform(data_scaled)
```

6. Feasibility Assessment

6.1 EDA Results

Exploratory Data Analysis (EDA) revealed key insights:

```
# Correlation heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```

- Positive correlation between urbanization and literacy rates.
- Significant improvement in literacy rates between 2001 and 2011.

6.2 Metrics for Future Evaluation

Metrics for evaluating future models include:

- **Accuracy**: To measure prediction reliability.
- **F1 Score**: For balanced evaluation of precision and recall.

7. Conclusion

This report highlights the critical role of data cleaning and visualization in deriving actionable insights. The application of advanced analytical techniques ensures that datasets are accurate and reliable for decision-making.By utilizing tools like Matplotlib, Seaborn, and Tableau, data trends and patterns were clearly identified.

Lessons Learned

- The importance of thorough data cleaning to ensure reliable analysis.
- The value of visualizations in uncovering trends and insights.

Next Steps

- Implement advanced AI techniques for predictive modeling.
- Expand the dataset to include additional years and demographic factors.
- Develop interactive dashboards for real-time insights.