Phase 2 - Solution Architecture

Data Exploration and Solution Planning

College Name: Maratha Mandal Engineering

College Group Members:

- Sumit Kalal (CAN_33834392): Led data exploration and cleaning, including handling missing values and visualizing inconsistencies.
- Prathamesh Patil (CAN_34003622): Conducted feature engineering, dimensionality reduction, and aligned insights with business objectives.
- Gireesh A Tallur (CAN_33834641): Researched anomaly detection models, implemented Isolation Forest, and validated model performance.
- Rohil Uday Gurav (CAN_33986080): Coordinated the project, enhanced visualizations, integrated pipelines, optimized model deployment, and prepared reports.

1. Overview of Data Visualization and Analysis

Phase 1 documented progress in understanding CRM data quality issues, identifying inconsistencies, and planning for an AI-driven solution. This phase focuses entirely on exploratory data analysis (EDA) and leveraging visualizations to refine our understanding and guide model design before training any models.

Objectives:

- Develop visualizations to identify CRM data trends and inconsistencies.
- Gain actionable insights through EDA to guide model selection and preprocessing.
- Establish hypotheses and criteria for detecting and correcting data inaccuracies.

2.Data Cleaning and Preparation

2.1 Handling Missing Values

Using CRM data, missing values were identified and addressed as follows:

- **Numerical Features:** Imputed using the median to mitigate outlier effects.
- **Categorical Features:** Assigned a placeholder value "Unknown" for missing categories, ensuring data preservation.

Code Example:

python

Copy code

```
import pandas as pd


# Load the dataset

crm_data = pd.read_csv("crm_data.csv")


# Impute numerical columns

numerical_cols = crm_data.select_dtypes(include=['float64', 'int64']).columns

crm_data[numerical_cols] =
crm_data[numerical_cols].fillna(crm_data[numerical_cols].median())


# Impute categorical columns

categorical_cols = crm_data.select_dtypes(include=['object']).columns

crm_data[categorical_cols] = crm_data[categorical_cols].fillna('Unknown')
```

2.2 Managing Outliers

Outliers in CRM data (e.g., erroneous sales or demographic entries) were visualized and treated:

- **Detection:** Identified using boxplots and Z-score analysis.
- **Treatment:**
  - Winsorization: Capped extreme values within the 99th percentile.
  - Exclusion: Removed invalid entries (e.g., negative values).

Code Example:

python

Copy code

```
import numpy as np


# Capping extreme values
crm_data['CustomerValue'] = np.clip(crm_data['CustomerValue'],
                            crm_data['CustomerValue'].quantile(0.01),
                            crm_data['CustomerValue'].quantile(0.99))


# Removing corrupted rows
crm_data = crm_data[crm_data['CustomerValue'] > 0]
```

2.3 Resolving Duplicates and Inconsistencies

- **Duplicates:** Flagged and merged duplicate customer profiles.
- **Inconsistencies:** Addressed conflicts in contact details and demographics.

Code Example:

python

Copy code

```
# Removing duplicates
crm_data = crm_data.drop_duplicates()
```

---

3. Data Visualization

3.1 Tools for Visualization

To facilitate effective data analysis, the following Python libraries were employed:

- **Matplotlib:** For static trend visualizations.
- **Seaborn:** For detailed correlation heatmaps.
- **Plotly:** For interactive anomaly exploration.

3.2 Key Visualizations and Insights

- **Time-Series Analysis:** Visualized data entry trends over time.

- **Correlation Heatmap:** Highlighted relationships between CRM attributes like sales and engagement.
- **Scatterplots:** Clusters of anomalies were visualized for further analysis.
- **Boxplots:** Outliers in numerical features like revenue were identified.

Example Code for Visualizations:

python

Copy code

```
import matplotlib.pyplot as plt

import seaborn as sns


# Correlation heatmap

plt.figure(figsize=(10, 8))

sns.heatmap(crm_data.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.show()


# Boxplot for customer value

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

sns.boxplot(crm_data['CustomerValue'])

plt.title("Customer Value Distribution")

plt.show()
```

---

4. Model Research and Selection Rationale

4.1 Research into Techniques

Based on CRM data characteristics, the following techniques were evaluated:

1. **Isolation Forest:** Effective for identifying inconsistent or erroneous entries.
2. **Logistic Regression:** Simple and interpretable, useful for detecting data patterns.
3. **Random Forest:** Suitable for flagging duplicate or incomplete records.

Justification for Isolation Forest:

- **Robustness:** Efficient with imbalanced data.
- **Scalability:** Handles high-dimensional datasets effectively.
- **Interpretability:** Provides anomaly scores for each entry.

---

5. Data Transformation and Feature Engineering

5.1 Feature Scaling

- **Standardization:** Applied to numerical attributes for uniformity.
- **Min-Max Scaling:** Used for features with skewed distributions.

Code Example:

python

Copy code

```
from sklearn.preprocessing import StandardScaler, MinMaxScaler


# Standardization
scaler = StandardScaler()
crm_data['StandardizedValue'] = scaler.fit_transform(crm_data[['CustomerValue']])


# Min-Max Scaling
crm_data['ScaledFeature'] = MinMaxScaler().fit_transform(crm_data[['SomeFeature']])
```

5.2 Encoding Categorical Variables

- **One-Hot Encoding:** Preserved interpretability by creating binary features for categorical attributes.

Code Example:

python

Copy code

```
# One-Hot Encoding
crm_data_encoded = pd.get_dummies(crm_data, columns=['CategoryFeature'])
```

5.3 Dimensionality Reduction

- **PCA:** Reduced dimensionality while retaining 95% variance.

Code Example:

python

Copy code

```
from sklearn.decomposition import PCA


pca = PCA(n_components=5)

data_pca = pca.fit_transform(crm_data.drop(columns=['Target']))
```

---

6. Feasibility Assessment

6.1 EDA Results

- **Hypotheses:** Formed based on observed data patterns.
- **Algorithm Testing:** Initial anomaly scoring simulated with Isolation Forest.
- **Business Alignment:** Results aligned with CRM improvement objectives.

6.2 Metrics for Future Evaluation

- **Precision and Recall:** To balance false positives and negatives.
- **ROC-AUC:** Evaluates sensitivity-specificity trade-offs.

---

7. Conclusion

Phase 2 reinforced the foundation established in Phase 1 by enhancing data understanding through EDA and visualization. Model evaluations suggested Isolation Forest as the most suitable choice for CRM data anomaly detection.

Next Steps:

- Train and validate selected models.
- Develop APIs for integrating models with CRM systems.
- Create dashboards for monitoring data quality.