Phase:03 DEVELOPMENT

Project-02:AI-Enhanced Data Accuracy in CRM Systems Using AI Data Analysts

College Name:Dr.Ambedkar Institute Technology(Dr.AIT).

RAVI K R **CAN_36049468**

AI-Enhanced Data Accuracy in CRM Systems Using AI Data Analysts

Project Development:

Phase 1: Requirement Analysis

- **Goal**: Improve CRM data accuracy by cleaning, deduplicating, and segmenting customer records using AI techniques.
- **Stakeholders**: Data analysts, CRM admins, marketing & sales teams.

Tools & Libraries:

- Python, Pandas, NumPy
- Scikit-learn, FuzzyWuzzy
- Matplotlib/Seaborn (for data visualization)
- Jupyter Notebook or VS Code

Phase 2: Dataset Preparation

Step 1: Import Libraries

```python
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans, DBSCAN
from fuzzywuzzy import fuzz
import matplotlib.pyplot as plt
import seaborn as sns
```

Step 2: Load CRM Data

```python
df = pd.read_csv("crm_dataset.csv")
print(df.head())
```

Phase 3: Data Cleaning

Step 1: Handle Missing Values

```python
# Visualize missing values
sns.heatmap(df.isnull(), cbar=False)
df.fillna(method='ffill', inplace=True)
```

Step 2: Standardize and Normalize

```python
scaler = StandardScaler()
df[['Age', 'Income']] = scaler.fit_transform(df[['Age', 'Income']])
```

Step 3: Format Strings

```python
df['Email'] = df['Email'].str.lower().str.strip()
df['Name'] = df['Name'].str.title()
```

Phase 4: Duplicate Detection Using AI

Approach A: Fuzzy String Matching

```python
duplicates = []
for i in range(len(df)):
    for j in range(i+1, len(df)):
        similarity = fuzz.token_sort_ratio(df.loc[i, 'Name'], df.loc[j, 'Name'])
        if similarity > 90:
            duplicates.append((df.loc[i, 'Name'], df.loc[j, 'Name'], similarity))

print("Duplicate Pairs:", duplicates[:5])
```

Approach B: Clustering with DBSCAN for Entity Matching

```
db = DBSCAN(eps=0.5, min_samples=2).fit(df[['Age', 'Income']])
df['Cluster'] = db.labels_
```

Phase 5: Data Validation Using Supervised Learning

Train a Model to Predict Record Validity

```
df['Valid'] = np.random.randint(0, 2, size=len(df))  # Sample validity label
X = df[['Age', 'Income']]
y = df['Valid']

model = RandomForestClassifier()
model.fit(X, y)

# Predict probability of accuracy
df['Accuracy_Prob'] = model.predict_proba(X)[:, 1]
```

Phase 6: Customer Segmentation

Apply K-Means Clustering

```
kmeans = KMeans(n_clusters=3, random_state=0)
df['Segment'] = kmeans.fit_predict(df[['Age', 'Income']])

# Visualize Segments
plt.figure(figsize=(8,5))
sns.scatterplot(data=df, x='Age', y='Income', hue='Segment', palette='Set2')
plt.title("Customer Segments based on Age and Income")
plt.show()
```

Phase 7: Report Generation

Create Summary Report

```
report = df.groupby('Segment')[['Age', 'Income', 'Accuracy_Prob']].mean()
print(report)

# Save cleaned & segmented data
df.to_csv("cleaned_crm_data.csv", index=False)
```

Final Output

- Cleaned CRM dataset (`cleaned_crm_data.csv`)
- Accuracy prediction for each record
- Customer segments for marketing analysis
- Duplicate name reports

Project Directory Structure

```
crm_ai_cleaner/
│
├── data/
│   └── crm_dataset.csv
│
├── scripts/
│   └── clean_validate_segment.py
│
├── output/
│   ├── cleaned_crm_data.csv
│   └── duplicates_report.csv
│
├── visuals/
│   └── segments_plot.png
│
└── README.md
```