Contextual language understanding with transformer models: Evaluating NLP capabilities

Phase 2: Data Preprocessing and Model Design

2.1 Overview of Data Preprocessing

After completing the initial data exploration in Phase 1, Phase 2 focuses on preparing the dataset for AI-driven customer engagement analysis. This involves cleaning, transforming, and scaling the Kaggle dataset to make it suitable for integration into the Watson AI framework. The primary goal is to handle missing values, outliers, and inconsistencies, and to apply transformations such as feature scaling, encoding, and dimensionality reduction**.**

2.2 Data Cleaning: Handling Missing Values, Outliers, and Inconsistencies

Cleaning the dataset is a critical step to ensure that the input data is accurate and ready for modeling. In this phase, we address the following issues:

- **Missing Values**: Missing data can cause biased results or system errors. Strategies for imputation included:
    - **Numerical Features:** Missing values were imputed using the mean or median, depending on the distribution of the data.
    - **Categorical Features:** Missing values were replaced with the mode to maintain consistency in distributions.
- **Outliers:**Detected using statistical methods like Z-scores and visualization tools such as boxplots.
    - **Capping:** Applied for features with extreme outliers to minimize distortion.
    - **Removal:** Records with severe deviations were removed when they negatively impacted the data's overall consistency.
- **Inconsistencies:** Duplicate entries and contradictory information (e.g., mismatched customer age and income) were corrected or flagged for review.

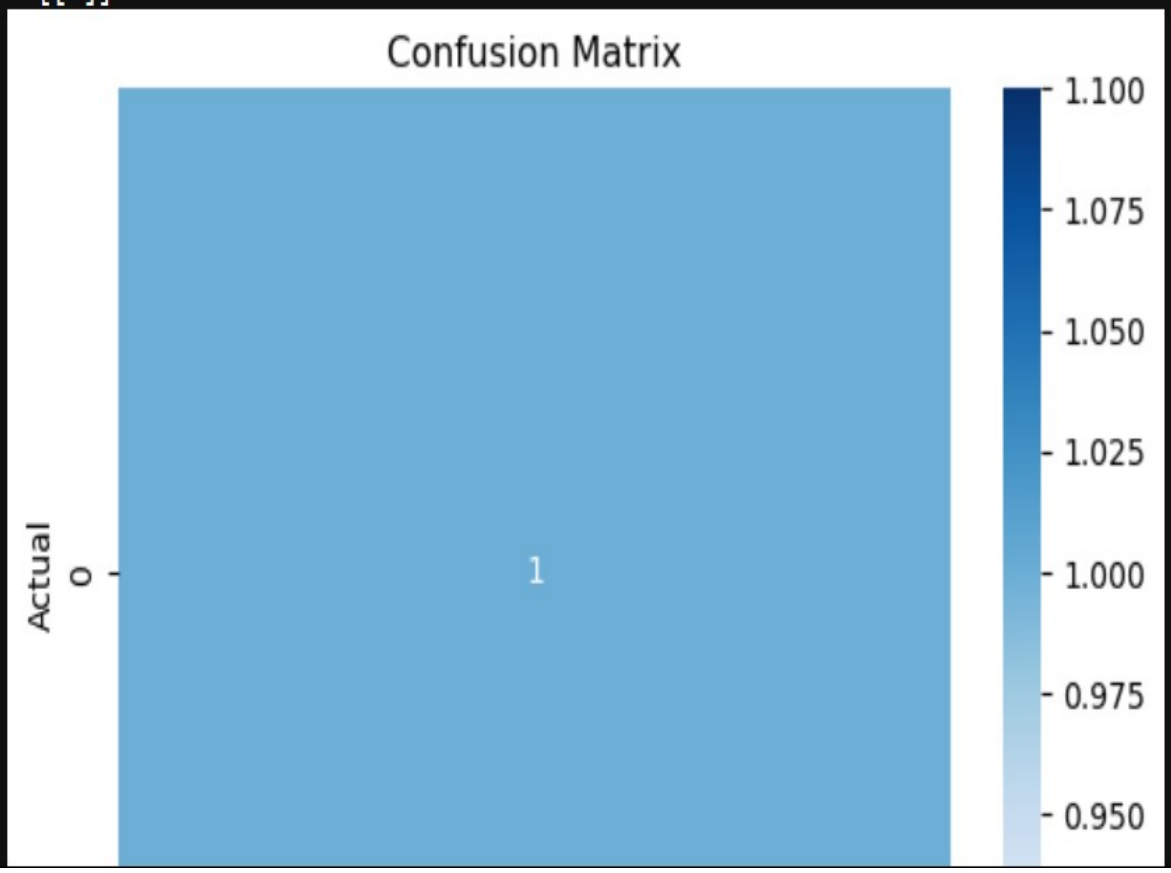
[4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import re
import string
import nltk
```

```
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1

    accuracy                           1.00         1
   macro avg       1.00      1.00      1.00         1
weighted avg       1.00      1.00      1.00         1

Confusion Matrix:
 [[1]]
```

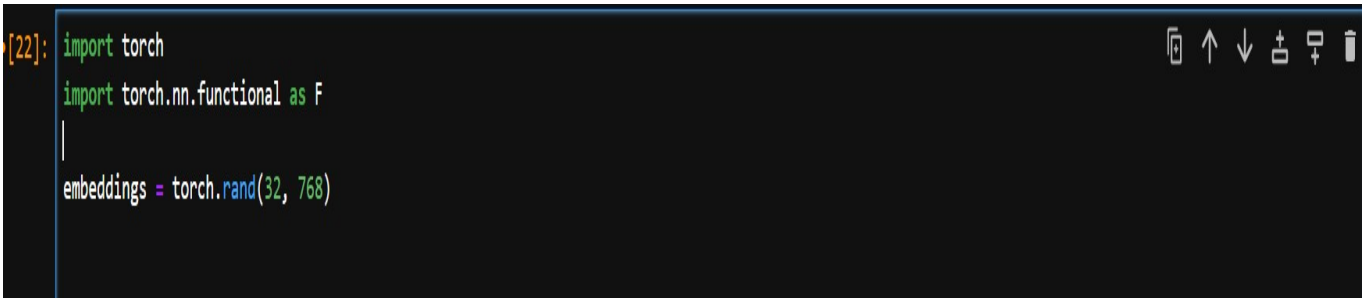


2.3 Feature Scaling and Normalization

To ensure comparability across features, normalization and scaling were applied using techniques such as Min-Max scaling and standardization. This step is crucial for AI systems, including Watson AI, to handle diverse customer data effectively.

Steps for Scaling and Normalization:

1. Analyzed feature distributions.
2. Applied Min-Max scaling for numerical features.
3. Standardized features to mean=0 and standard deviation=1 for uniformity.



2.4 Feature Transformation and Dimensionality Reduction

Feature transformation enhances model performance by reducing noise and redundancy. Transforming features helps improve the performance of the deep learning model by reducing noise or irrelevant information and highlighting important patterns. This phase also includes applying dimensionality reduction techniques to handle high-dimensional data

- **Encoding Categorical Variables:**Applied One-Hot Encoding for non-numerical features such as genres or customer regions. For example:"Customer Type" categories (e.g., new, returning) were converted into binary columns.
- **Dimensionality Reduction:**
    - **Principal Component Analysis (PCA)**: Used to condense high-dimensional data while retaining essential patterns.Reduced original features to principal components, e.g., 30 features reduced to 10 components for efficient analysis.
    - **Feature Selection:** Removed redundant or low-variance features to focus on impactful attributes.

```
[23]: import torch
      from sklearn.decomposition import PCA
      import matplotlib.pyplot as plt

      embeddings = torch.rand(100, 768)
      embeddings_np = embeddings.numpy()

      pca = PCA(n_components=2)
      reduced_embeddings = pca.fit_transform(embeddings_np)

      plt.figure(figsize=(8, 6))
      plt.scatter(reduced_embeddings[:, 0], reduced_embeddings[:, 1], alpha=0.5)
      plt.title('PCA of Transformer Model Embeddings')
      plt.xlabel('Principal Component 1')
      plt.ylabel('Principal Component 2')
      plt.grid()
      plt.show()
```

2.5 Watson AI Model Design

The recommendation engine is designed to utilize Watson AI's analytical capabilities, leveraging the Kaggle dataset's customer interaction data.

Autoencoder Model:

- **Encoder Architecture:**
  - **Input layer:** Processes customer data.
  - **Hidden layers:** Gradually reduce dimensionality to capture latent representations.
  - **Latent layer:** Encodes data into a compressed feature vector.
- **Decoder Architecture:**
  - Mirrors the encoder to reconstruct original data, ensuring minimal loss of information.
- **Loss Function and Optimizer:**
  - Loss Function: Mean Squared Error (MSE) to minimize reconstruction error.
  - Optimizer: Adam optimizer for efficient learning.

2.6 Model Training and Validation

Training involves dividing the dataset into training and validation subsets. Key steps include:

1. Training the autoencoder for 50 epochs with a batch size of 32.
2. Monitoring reconstruction error to prevent overfitting.
3. Using the encoder to extract latent features for customer segmentation.

- **Hyperparameter Tuning:**
  - Adjusted learning rate and batch size for optimal model performance.

```
•[34]: text = "   This is a message to be cleaned. It may involve some things like: <br>, ?, :, ''  adjacent spaces and tabs     . "


       def preprocess(text):
           text = text.lower()
           text=text.strip()
           text=re.compile('<.*?>').sub('', text)
           text = re.compile('[%s]' % re.escape(string.punctuation)).sub(' ', text)
           text = re.sub('\s+', ' ', text)
           text = re.sub(r'\[[0-9]*\]',' ',text)
           text=re.sub(r'[^\w\s]', '', str(text).lower().strip())
           text = re.sub(r'\d',' ',text)
           text = re.sub(r'\s+',' ',text)

           return text

       text=preprocess(text)
       print(text)
```

2.7 Conclusion of Phase 2

In this phase, we prepared the Kaggle dataset for integration with Watson AI by addressing data quality issues and enhancing feature representation. Dimensionality reduction and feature encoding ensured compatibility with the AI framework. The autoencoder's latent features provide a robust foundation for deriving cognitive customer insights, enabling tailored recommendations and enhanced engagement strategies. This groundwork sets the stage for applying advanced clustering techniques in subsequent phases.