Assessment Report on

"MBTI Personality Type Prediction using NLP"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25 in

CSE(AI)-B

By

Group No. 7

Ashish Srivastava

Harshit Patel

Harsh

Divyanshu chaudhary

Under the supervision of

"Shivansh Prasad"

KIET Group of Institutions, Ghaziabad

1. Introduction

With the increasing availability of user-generated content on social media, analyzing personality traits using written text has become a fascinating research area in Natural Language Processing (NLP). This project focuses on predicting the MBTI (Myers-Briggs Type Indicator) personality types of individuals based on their writing style. Using a dataset of user posts and machine learning models, the project aims to automatically classify a user's MBTI type using NLP techniques.

---

2. Problem Statement

To classify a user's MBTI personality type (e.g., INTP, ENFJ, ISTJ) based on their written text using NLP and supervised machine learning models. This can be helpful for psychological analysis, personalized recommendations, and HR assessments.

---

3. Objectives

● Preprocess raw text data from social media posts for model training.

● Convert textual data into numerical format using TF-IDF.

● Train a Logistic Regression classifier to predict personality types.

● Evaluate model performance using standard classification metrics.

● Visualize prediction results using a confusion matrix heatmap.

---

4. Methodology

● **Data Collection:**

- The dataset was obtained from Kaggle, containing MBTI types and users' posts.

● **Data Preprocessing:**

- Removing URLs, punctuation, and stopwords.
- Converting text to lowercase and applying stemming.
- Combining multiple posts into one per user.

● **Feature Extraction:**

- Applying TF-IDF vectorization to convert text into feature vectors.

● **Model Building:**

- Encoding MBTI labels numerically (0–15).
- Splitting data into training and test sets.
- Training Logistic Regression on vectorized data.

● **Model Evaluation:**

- Calculating accuracy, precision, recall, and F1-score.
- Creating and visualizing a confusion matrix with Seaborn heatmap.

---

5. Data Preprocessing

The text data was cleaned and prepared using the following steps:

● Converted text to lowercase.

● Removed hyperlinks, special characters, and punctuation.

● Removed common English stopwords (like "the", "and", etc.).

● Applied stemming to reduce words to their root forms.

● Transformed cleaned text into numerical vectors using TF-IDF.

6. Model Implementation

Logistic Regression was chosen for its effectiveness in multi-class classification and interpretability. The model was trained on TF-IDF features extracted from the processed user posts and used to predict MBTI types on the test set.

7. Evaluation Metrics

The performance of the model was evaluated using:

- **Accuracy** – Correct predictions over total predictions.
- **Precision** – Proportion of correct positive predictions.
- **Recall** – Proportion of actual positives correctly predicted.
- **F1 Score** – Harmonic mean of precision and recall.
- **Confusion Matrix** – Used to visualize classification results using a heatmap.

8. Results and Analysis

- The Logistic Regression model showed reasonable accuracy in predicting MBTI types.
- The confusion matrix showed how the model performed across 16 classes.
- Certain personality types were more frequently misclassified due to textual similarity.
- Precision and recall scores highlighted how well the model detected specific MBTI types.

9. Conclusion

This project demonstrates the potential of using NLP and machine learning to classify personality types based on written text. Despite being a simple baseline using Logistic Regression, the results were promising. Further improvements can include deep learning models like LSTM or BERT and addressing class imbalance in the dataset for better accuracy.

---

10. References

● scikit-learn documentation
● pandas documentation
● Seaborn visualization library
● Kaggle MBTI Dataset
● Research papers on MBTI and NLP

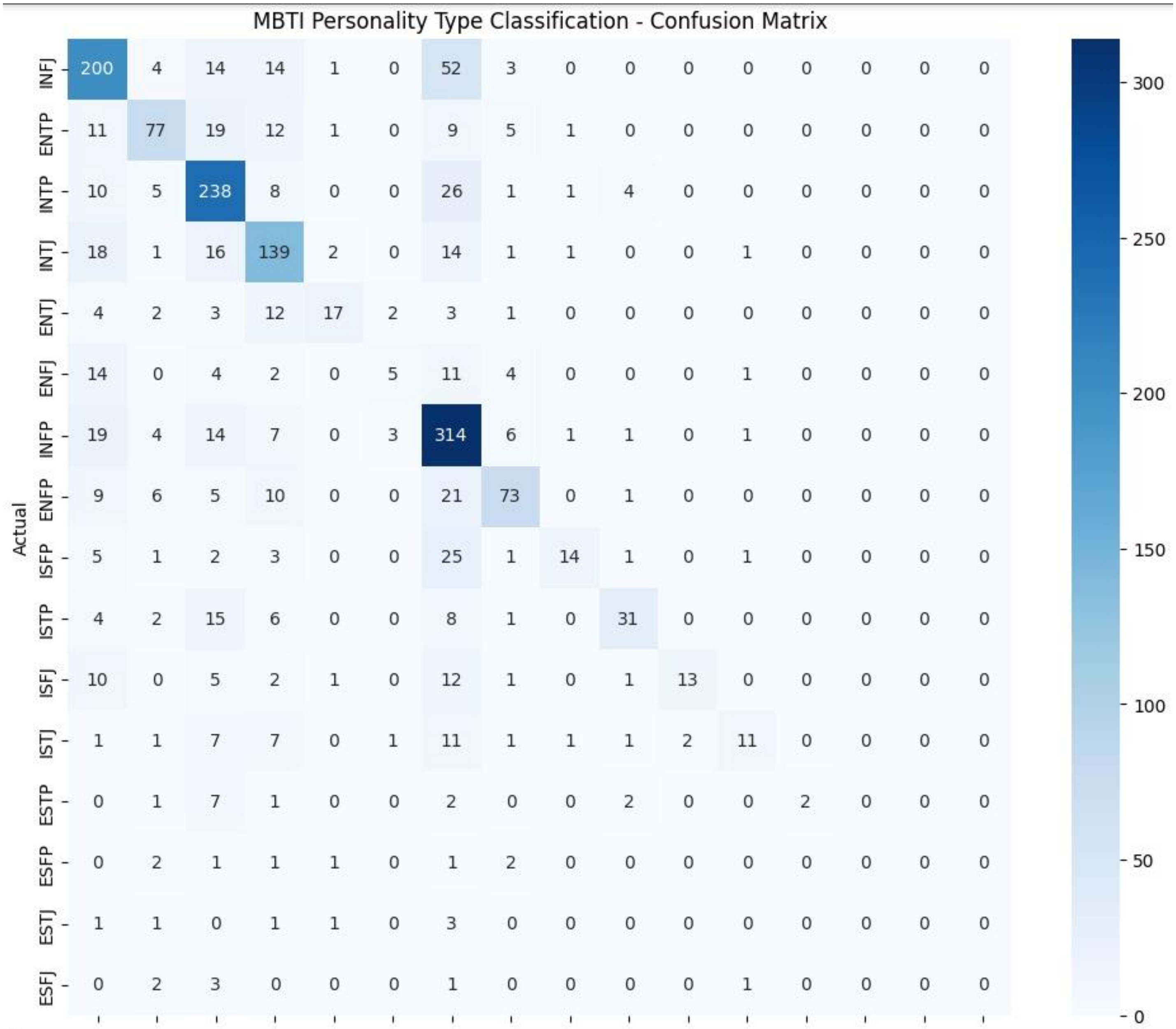
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```python
import pandas as pd
import numpy as np
import re
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("/content/mbti_1.csv")

# Inspect data
print(df.head())
print(df['type'].value_counts())

# Preprocessing function
def clean_text(text):
    text = text.lower()
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[^a-z\s]', ' ', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

df['clean_posts'] = df['posts'].apply(clean_text)

# Map personality types to numbers and reverse map
types = list(df['type'].unique())
type_to_id = {t: i for i, t in enumerate(types)}
id_to_type = {i: t for t, i in type_to_id.items()}
df['label'] = df['type'].map(type_to_id)

# Split data
X_train, X_test, y_train, y_test = train_test_split(
    df['clean_posts'], df['label'], test_size=0.2, random_state=42)

# TF-IDF Vectorizer with stopwords
vectorizer = TfidfVectorizer(max_features=5000, stop_words='english')
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

# Logistic Regression classifier
```

```python
# Logistic Regression classifier
clf = LogisticRegression(max_iter=300, solver='saga', n_jobs=-1, random_state=42)
clf.fit(X_train_tfidf, y_train)

# Evaluate on test set
y_pred = clf.predict(X_test_tfidf)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=types))

# Confusion matrix plot
conf_mat = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(12,10))
sns.heatmap(conf_mat, xticklabels=types, yticklabels=types, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('MBTI Personality Type Classification - Confusion Matrix')
plt.show()
```

```python
# Part 2: User input for prediction

def clean_text(text):
    import re
    text = text.lower()
    text = re.sub(r'http\S+', '', text)
    text = re.sub(r'[^a-z\s]', ' ', text)
    text = re.sub(r'\s+', ' ', text).strip()
    return text

while True:
    user_text = input("\nEnter text to predict MBTI personality type (or type 'exit' to quit): ")
    if user_text.lower() == 'exit':
        print("Exiting... Goodbye!")
        break
    cleaned_input = clean_text(user_text)
    input_tfidf = vectorizer.transform([cleaned_input])
    pred_label = clf.predict(input_tfidf)[0]
    pred_type = id_to_type[pred_label]
    print(f"Predicted MBTI personality type: {pred_type}")
```


MBTI Personality Type Classification - Confusion Matrix
Actual
INFJ 200 4 14 14 1 0 52 3 0 0 0 0 0 0 0 0
ENTP 11 77 19 12 1 0 9 5 1 0 0 0 0 0 0 0
INTP 10 5 238 8 0 0 26 1 1 4 0 0 0 0 0 0
INTJ 18 1 16 139 2 0 14 1 1 0 0 1 0 0 0 0
ENTJ 4 2 3 12 17 2 3 1 0 0 0 0 0 0 0 0
ENFJ 14 0 4 2 0 5 11 4 0 0 0 1 0 0 0 0
INFP 19 4 14 7 0 3 314 6 1 1 0 1 0 0 0 0
ENFP 9 6 5 10 0 0 21 73 0 1 0 0 0 0 0 0
ISFP 5 1 2 3 0 0 25 1 14 1 0 1 0 0 0 0
ISTP 4 2 15 6 0 0 8 1 0 31 0 0 0 0 0 0
ISFJ 10 0 5 2 1 0 12 1 0 1 13 0 0 0 0 0
ISTJ 1 1 7 7 0 1 11 1 1 1 2 11 0 0 0 0
ESTP 0 1 7 1 0 0 2 0 0 2 0 0 2 0 0 0
ESFP 0 2 1 1 1 0 1 2 0 0 0 0 0 0 0 0
ESTJ 1 1 0 1 1 0 3 0 0 0 0 0 0 0 0 0
ESFJ 0 2 3 0 0 0 1 0 0 0 0 1 0 0 0 0
300
250
200
150
100
50
0

```
    type                                              posts
0  INFJ  'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1  ENTP  'I'm finding the lack of me in these posts ver...
2  INTP  'Good one  _____   https://www.youtube.com/wat...
3  INTJ  'Dear INTP,   I enjoyed our conversation the o...
4  ENTJ  'You're fired.|||That's another silly misconce...
type
INFP    1832
INFJ    1470
INTP    1304
INTJ    1091
ENTP     685
ENFP     675
ISTP     337
ISFP     271
ENTJ     231
ISTJ     205
ENFJ     190
ISFJ     166
ESTP      89
ESFP      48
ESFJ      42
ESTJ      39
Name: count, dtype: int64
Accuracy: 0.6536023054755044
```

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

        INFJ       0.65      0.69      0.67       288
        ENTP       0.71      0.57      0.63       135
        INTP       0.67      0.81      0.74       293
        INTJ       0.62      0.72      0.67       193
        ENTJ       0.71      0.39      0.50        44
        ENFJ       0.45      0.12      0.19        41
        INFP       0.61      0.85      0.71       370
        ENFP       0.73      0.58      0.65       125
        ISFP       0.74      0.26      0.39        53
        ISTP       0.74      0.46      0.57        67
        ISFJ       0.87      0.29      0.43        45
        ISTJ       0.69      0.25      0.37        44
        ESTP       1.00      0.13      0.24        15
        ESFP       0.00      0.00      0.00         8
        ESTJ       0.00      0.00      0.00         7
        ESFJ       0.00      0.00      0.00         7

    accuracy                           0.65      1735
   macro avg       0.57      0.38      0.42      1735
weighted avg       0.66      0.65      0.63      1735
```