In [1]:
```python
# Install necessary libraries
!pip install xgboost scikit-learn pandas

# Import Libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split, StratifiedKFold, GridSearchCV
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.impute import SimpleImputer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import f1_score
from xgboost import XGBClassifier

# Load Data
examples_df = pd.read_csv("examples.csv")
test_df = pd.read_csv("test.csv")

# Separate features and target label in examples data
X = examples_df.drop(columns=['Id', 'label'])
y = examples_df['label']

# Encode labels for XGBoost
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

# Define numerical and text columns
num_cols = [f'q{i}' for i in range(1, 21)]
text_cols = ['q21', 'q22']

# Fill missing values in text columns with empty strings for TF-IDF
X[text_cols] = X[text_cols].fillna("")

# Fill missing values in numerical columns
imputer = SimpleImputer(strategy="median")
X[num_cols] = imputer.fit_transform(X[num_cols])

# Train-test split
X_train, X_val, y_train, y_val = train_test_split(X, y_encoded, test_size=0.2, stratify=y, random_state=42

# Preprocess text features
tfidf_q21 = TfidfVectorizer(max_features=100, stop_words='english', ngram_range=(1, 2))
tfidf_q22 = TfidfVectorizer(max_features=100, stop_words='english', ngram_range=(1, 2))

X_train_q21 = tfidf_q21.fit_transform(X_train['q21']).toarray()
X_val_q21 = tfidf_q21.transform(X_val['q21']).toarray()

X_train_q22 = tfidf_q22.fit_transform(X_train['q22']).toarray()
X_val_q22 = tfidf_q22.transform(X_val['q22']).toarray()

# Concatenate TF-IDF features
X_train = np.hstack([X_train.drop(columns=text_cols).values, X_train_q21, X_train_q22])
X_val = np.hstack([X_val.drop(columns=text_cols).values, X_val_q21, X_val_q22])

# Standardize numerical features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)

# Define XGBoost model with parameter tuning
xgb_model = XGBClassifier(
    n_estimators=200,
    max_depth=6,
    learning_rate=0.05,
    subsample=0.8,
    colsample_bytree=0.8,
    random_state=42,
    eval_metric="mlogloss"
)

# Use cross-validation for a more robust score
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
grid_search = GridSearchCV(
    xgb_model,
    param_grid={
        'max_depth': [5, 6, 7],
        'learning_rate': [0.03, 0.05, 0.1],
        'n_estimators': [100, 200, 300]
    },
    scoring='f1_weighted',
    cv=cv,
    n_jobs=-1,
    verbose=1
)

# Fit the model with grid search
grid_search.fit(X_train, y_train)

# Best model and evaluation
```

```python
best_model = grid_search.best_estimator_
y_val_pred = best_model.predict(X_val)
f1 = f1_score(y_val, y_val_pred, average='weighted')
print("Optimized F1 Score with XGBoost:", f1)

# Process Test Data for Final Predictions
test_df[text_cols] = test_df[text_cols].fillna("")
test_df[num_cols] = imputer.transform(test_df[num_cols])

# Apply TF-IDF to test data
test_q21 = tfidf_q21.transform(test_df['q21']).toarray()
test_q22 = tfidf_q22.transform(test_df['q22']).toarray()
test_combined = np.hstack([test_df.drop(columns=['Id', 'q21', 'q22']).values, test_q21, test_q22])

# Standardize test data
test_combined = scaler.transform(test_combined)

# Predict on Test Data
predictions_encoded = best_model.predict(test_combined)
predictions = label_encoder.inverse_transform(predictions_encoded)

# Save submission
submission = pd.DataFrame({'Id': test_df['Id'], 'Label': predictions})
submission.to_csv("submission.csv", index=False)
```

```
Requirement already satisfied: xgboost in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (2.1.2)
Requirement already satisfied: scikit-learn in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (1.5.
2)
Requirement already satisfied: pandas in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (2.2.3)
Requirement already satisfied: numpy in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (from xgboos
t) (1.26.4)
Requirement already satisfied: scipy in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (from xgboos
t) (1.14.1)
Requirement already satisfied: joblib>=1.2.0 in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (from
scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /opt/anaconda3/envs/py3k/lib/python3.12/site-package
s (from scikit-learn) (3.5.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/anaconda3/envs/py3k/lib/python3.12/site-packa
ges (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (from
pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (fro
m pandas) (2023.3)
Requirement already satisfied: six>=1.5 in /opt/anaconda3/envs/py3k/lib/python3.12/site-packages (from pyth
on-dateutil>=2.8.2->pandas) (1.16.0)
Fitting 5 folds for each of 27 candidates, totalling 135 fits
Optimized F1 Score with XGBoost: 0.8999999999999998
```