

Enhanced Bug Report Classifier

User Manual

1. Introduction

The Enhanced Bug Report Classifier is a machine learning system designed to automatically identify performance-related bug reports in deep learning frameworks. This manual provides comprehensive instructions for using the tool effectively.

2. Basic Usage

Module Interface

```
from src.ensemble_classifier import EnhancedBugReportClassifier

# Initialize classifier
classifier = EnhancedBugReportClassifier()

# Train and evaluate
classifier.train(data)
metrics = classifier.evaluate(test_data)
```

Command Line Interface

```
python src/evaluate.py [--project PROJECT] [--n_iterations N]
```

Arguments:

- `--project`: Specific project to evaluate (default: all projects)
- `--n_iterations`: Number of evaluation iterations (default: 30)

Example:

```
python src/evaluate.py --project tensorflow --n_iterations 50
```

3. Advanced Usage

Custom Feature Engineering

```
# Add custom features
classifier.add_custom_features(feature_function)

# Modify feature weights
classifier.set_feature_weights(weights)
```

Ensemble Configuration

```
# Set base classifiers
classifier.set_base_classifiers([
    'naive_bayes',
    'random_forest',
    'xgboost',
    'lightgbm'
])

# Configure meta-classifier
classifier.set_meta_classifier('logistic_regression')
```

4. Configuration

Preprocessing Options

```
classifier.set_preprocessing_options({
    'remove_urls': True,
    'remove_numbers': False,
    'lowercase': True
})
```

Feature Extraction

```
classifier.set_feature_params({
    'max_features': 1000,
    'ngram_range': (1, 2),
    'min_df': 2
})
```

Classifier Settings

```
classifier.set_classifier_params('xgboost', {
    'n_estimators': 200,
    'learning_rate': 0.1
})
```

5. Best Practices

Data Preparation

- Clean and preprocess data thoroughly
- Validate input format
- Handle missing values appropriately

Model Training

- Use cross-validation
- Monitor training progress
- Save checkpoints regularly

Evaluation

- Use appropriate metrics
- Compare with baselines
- Document results

6. Troubleshooting

Memory Errors

- Reduce batch size
- Process smaller chunks of data
- Use smaller feature set

Performance Issues

- Enable parallel processing
- Use GPU acceleration
- Optimize feature extraction

Inconsistent Results

- Check random seed settings
- Verify data preprocessing
- Ensure consistent environment