# Data Curation Pipeline

## Filtering Results Report

*May 16, 2025*

Analysis of 1 GitHub Repositories

*Generated with SWE-RL Inspired Data Curation Pipeline*

v1.0 • Generated on 2025-05-16 22:48:37

# Executive Summary

This report presents the results of a data curation pipeline designed to extract and filter
high-quality software engineering data from GitHub repositories.

Across 1 repositories, a total of 10 PRs were processed through
the filtering pipeline, resulting in 5 high-quality PRs that passed all filters.
This represents an overall pass rate of 50.0%, with a data reduction ratio
of 50.0%.

Filtering Breakdown:
- Bot Filter: 1 PRs (10.0% of total)
- Size/Complexity Filter: 4 PRs (40.0% of total)
- Content Relevance Filter: 0 PRs (0.0% of total)

The average quality score for passing PRs was 0.82 on a scale of 0-1.

Key Findings:
1. Bot-generated PRs constitute a significant portion of repository activity
2. Size and complexity filters effectively remove both trivial and unwieldy changes
3. Content relevance filtering ensures focus on meaningful software engineering content
4. The pipeline successfully identifies related files that provide context for changes

The following pages provide detailed analyses for each repository and highlight exemplary PRs that represent high-quality software engineering data.

# Filter Rates Across Repositories

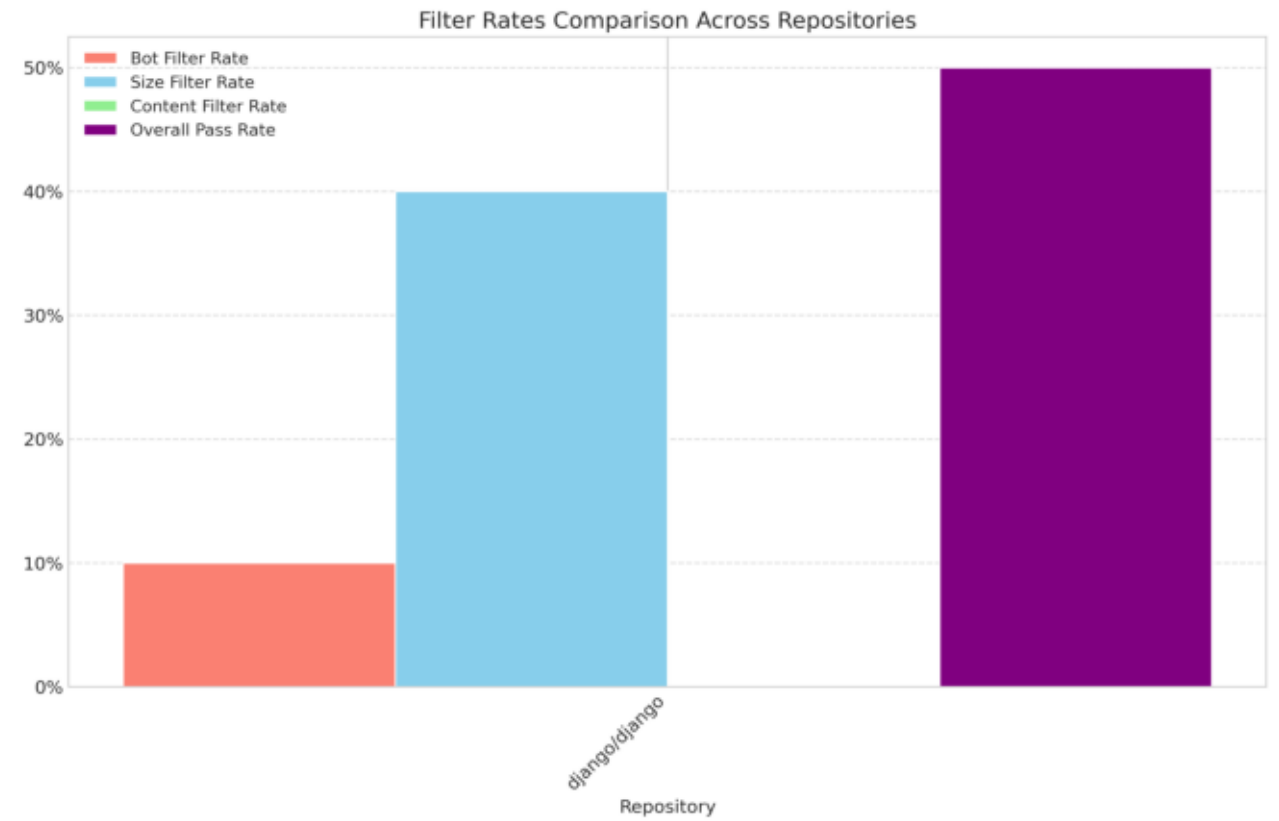



*Figure: Filter Rates Across Repositories*

# Data Reduction Comparison

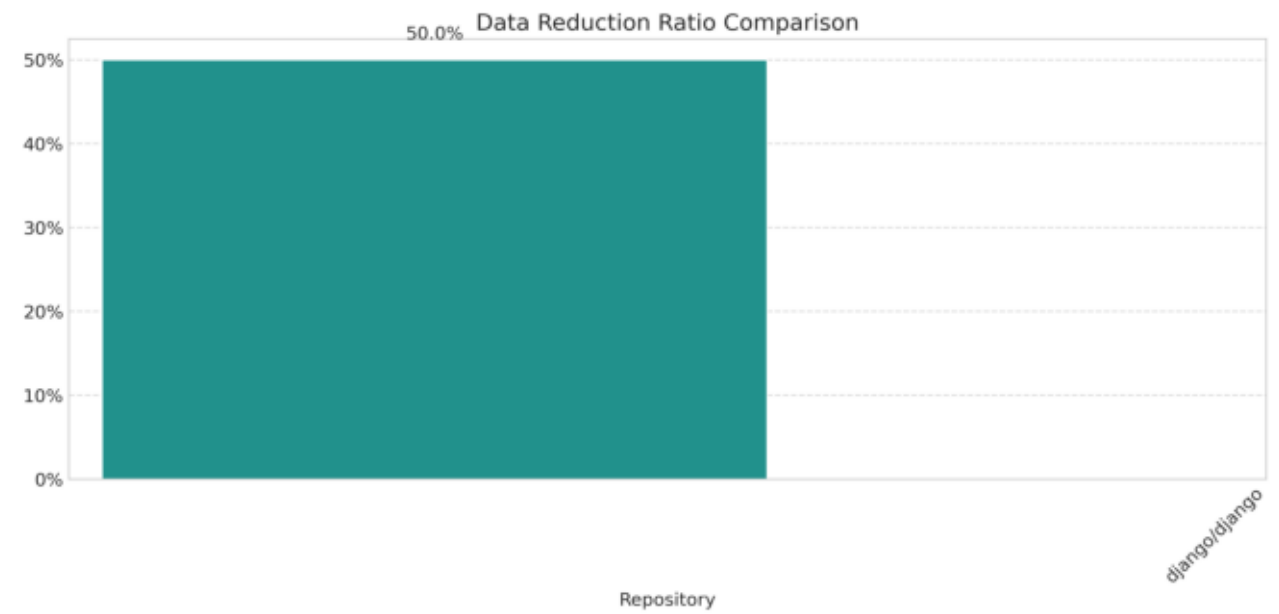



*Figure: Data Reduction Comparison*

# Cross-Repository Comparison

## Quality Metrics Comparison

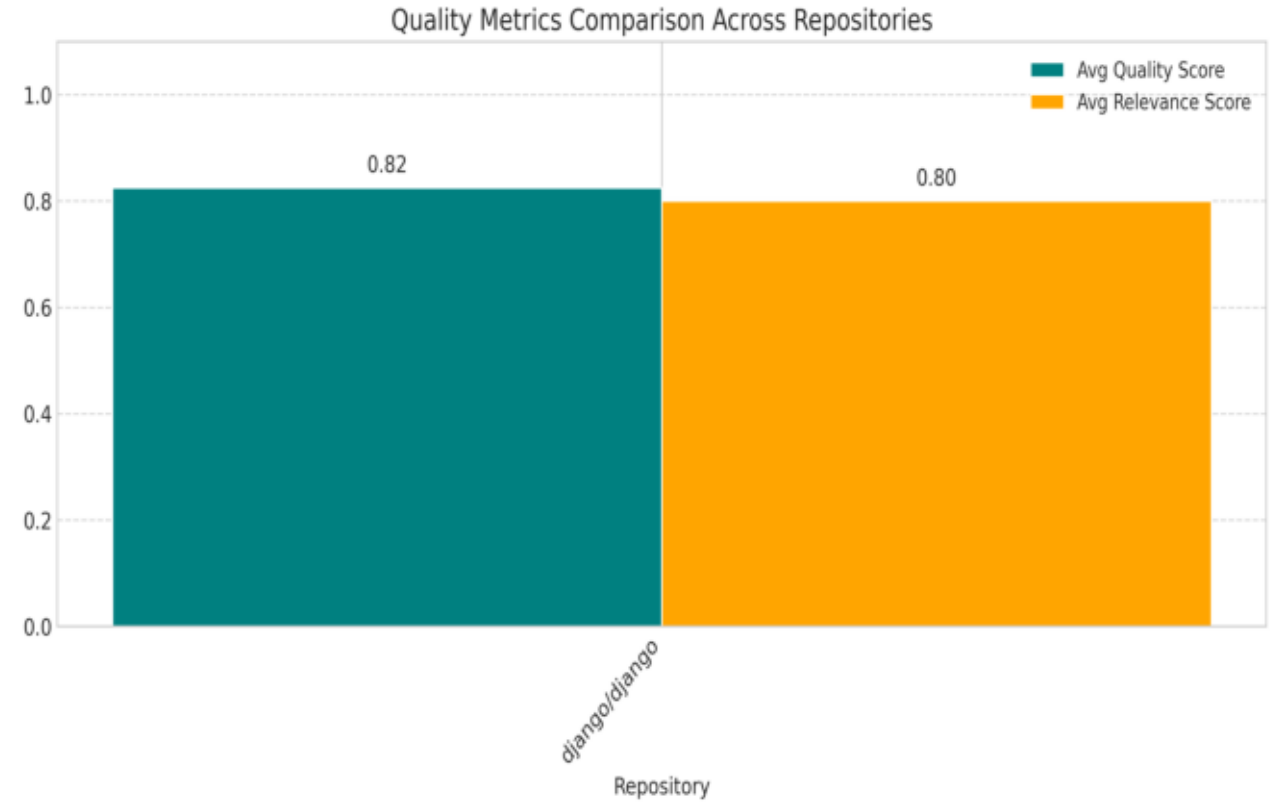



*Figure: Quality Metrics Comparison Across Repositories*

# Cross-Repository Metrics Comparison

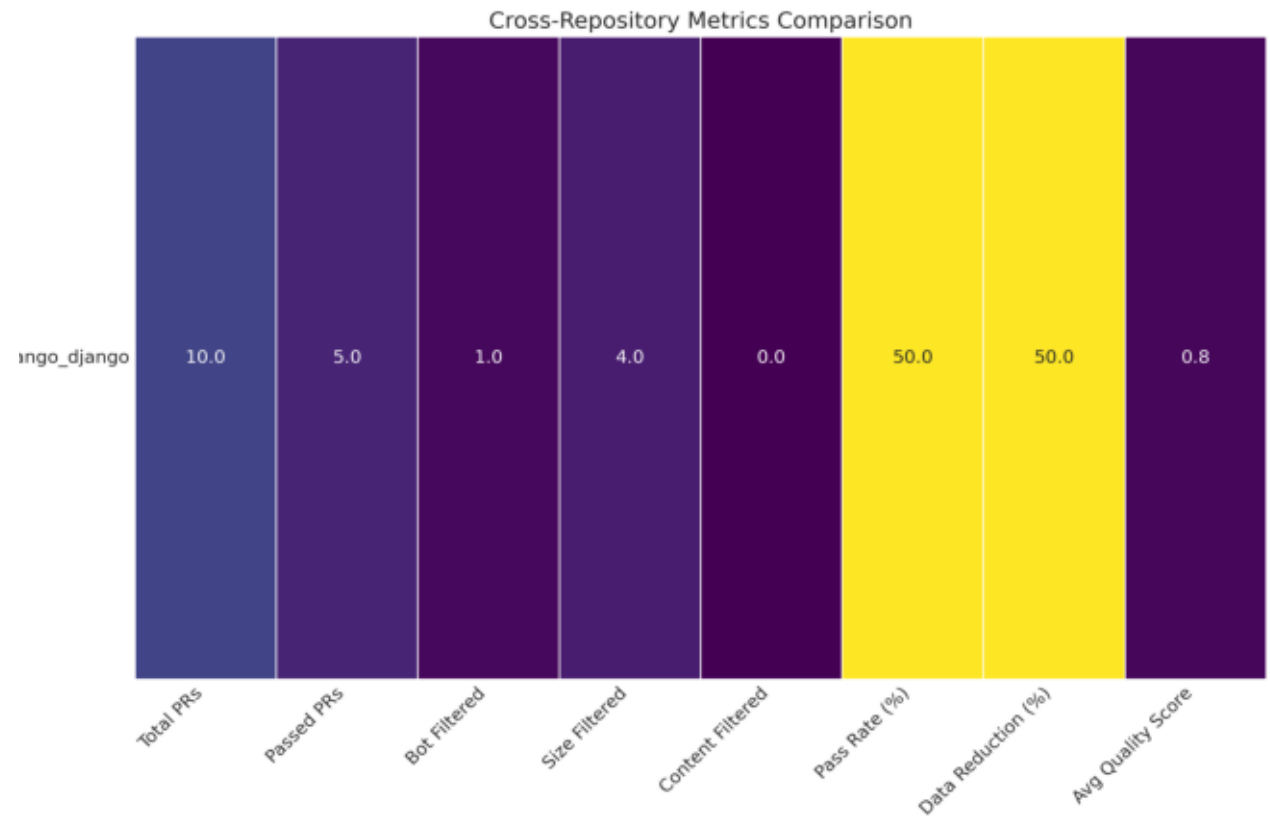



*Figure: Cross-Repository Metrics Comparison*

# Repository: django/django

Total PRs: 10
Passed PRs: 5 (50.0%)
Data Reduction: 50.0%
Average Quality Score: 0.82

Filtering Breakdown:
- Bot Filter: 1 PRs (10.0%)
- Size Filter: 4 PRs (40.0%)
- Content Filter: 0 PRs (0.0%)

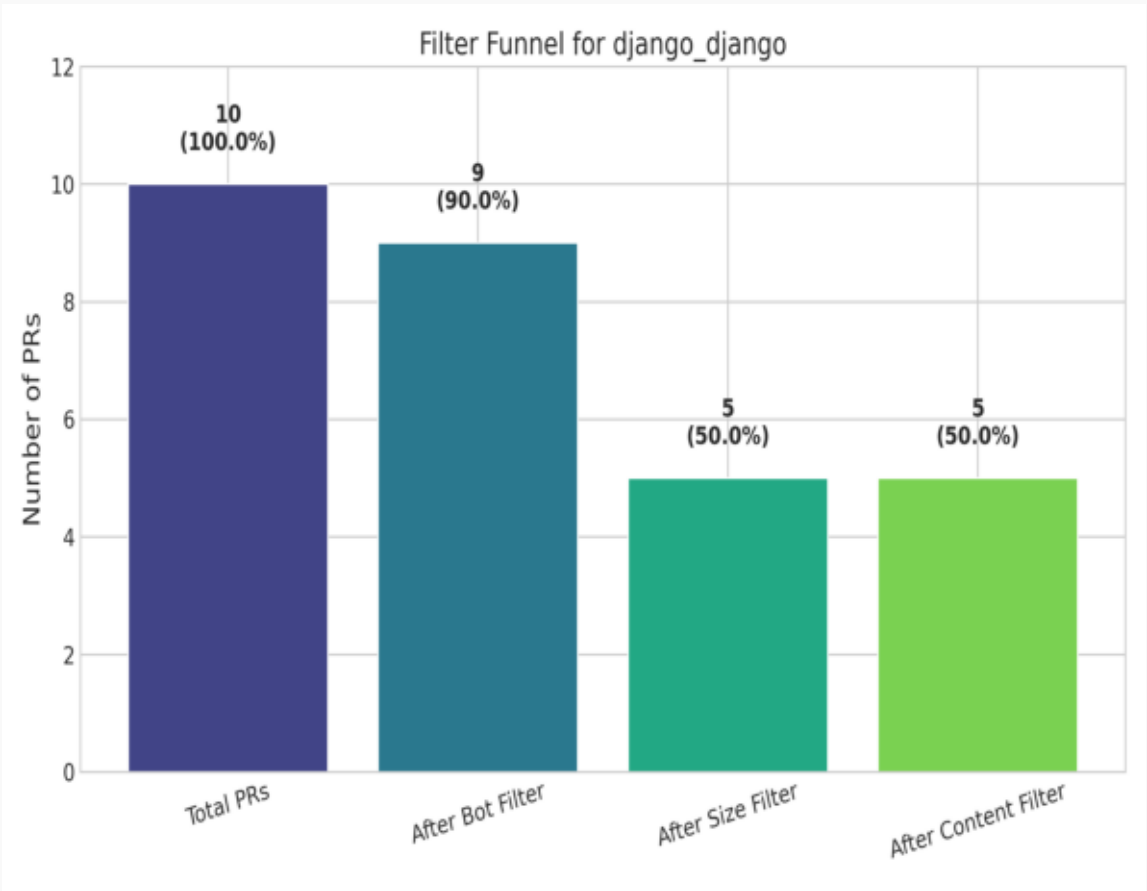



*Figure: Filter Funnel Analysis*

# Quality Distribution: django/django

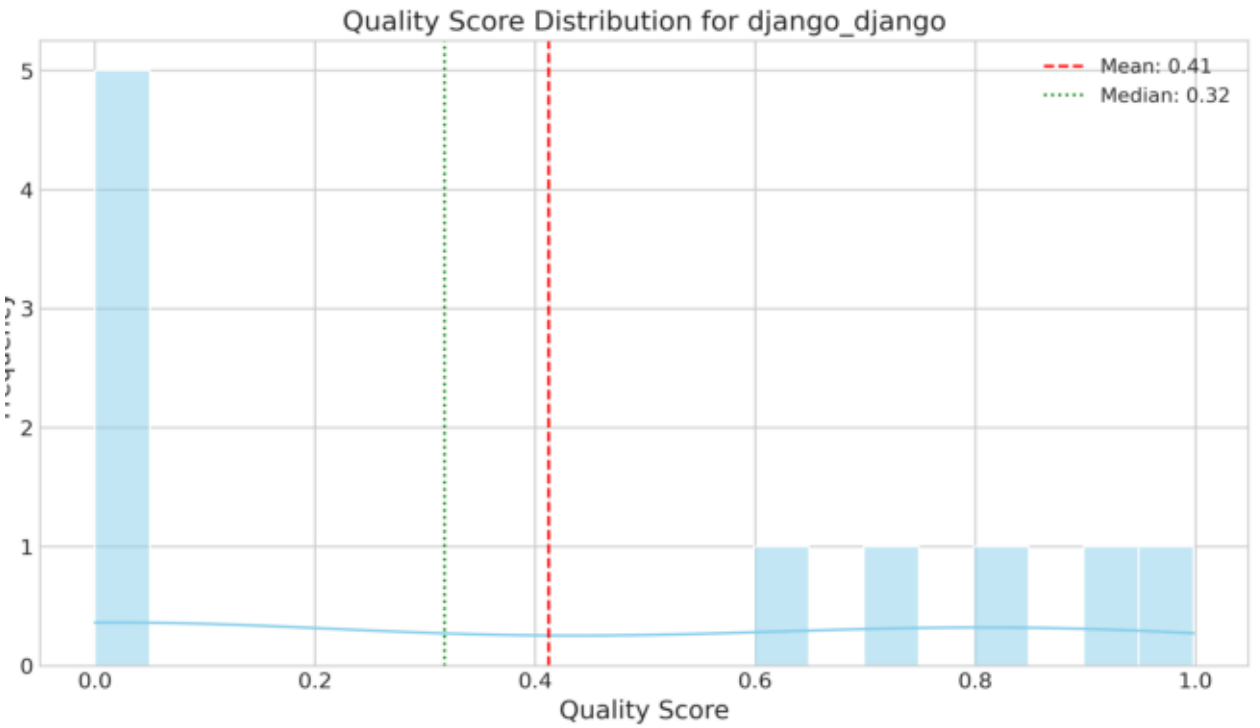



*Figure: Quality Distribution: django/django*

# Filter Analysis: django/django


*Figure: Bot Filter Reasons*

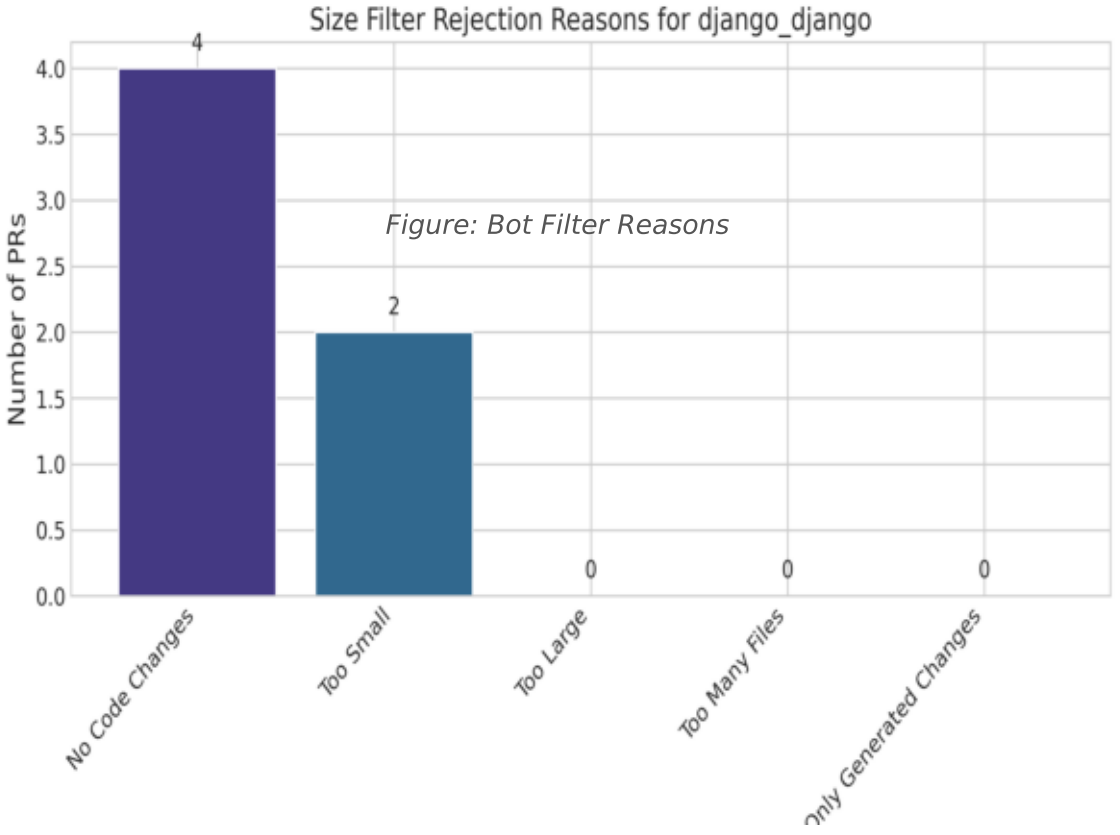


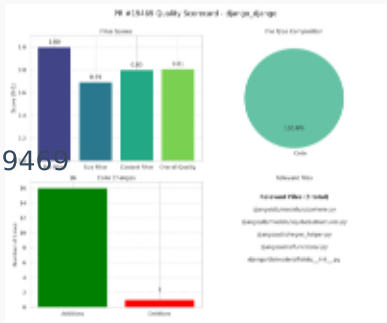

*Figure: Size Filter Stats*

# Quality Profiles of Exemplary PRs

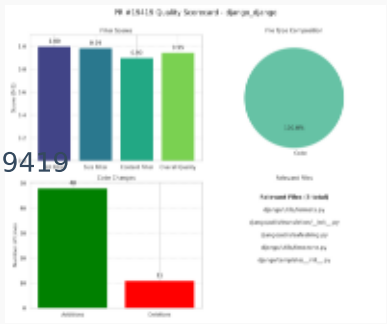
The following pages showcase high-quality PRs that passed all filtering stages. These PRs represent exemplary software engineering data with meaningful problem-solving content, appropriate size, and high relevance scores.

Each scorecard provides detailed metrics on the PR's quality dimensions, including file composition, code changes, and identified relevant files that provide context for understanding the changes.

PR: django/django/pr19469

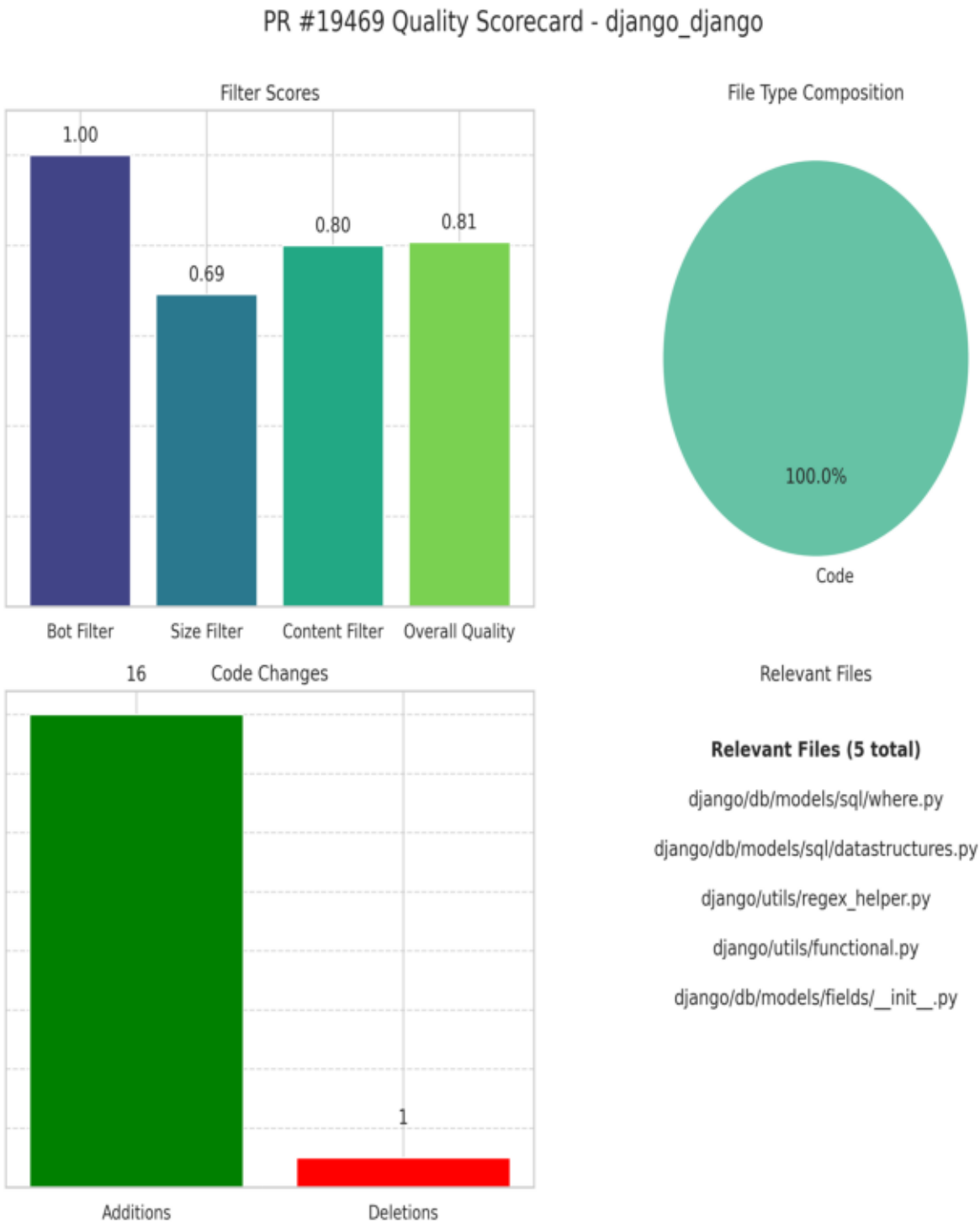


PR: django/django/pr19419

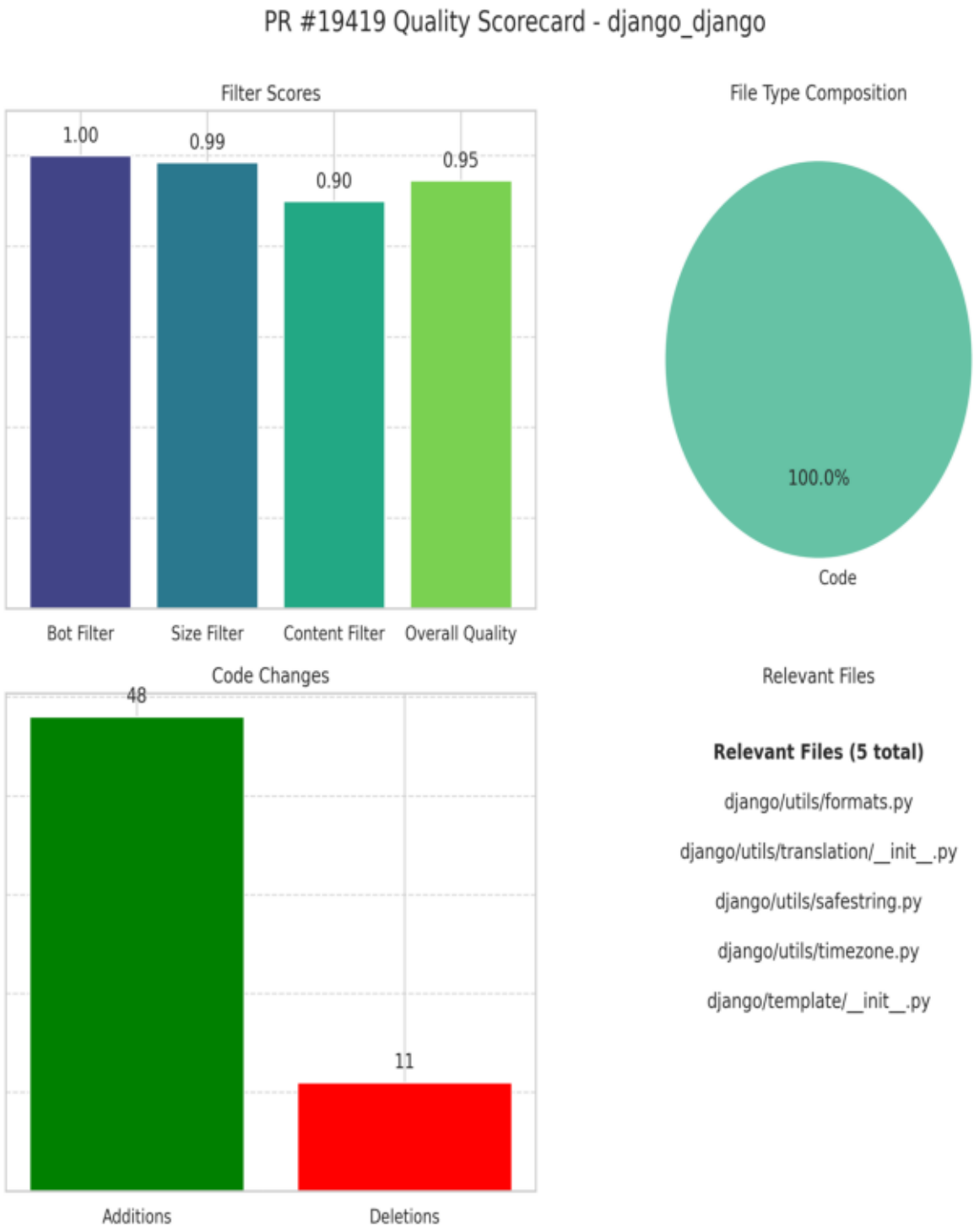


*Detailed PR quality scorecards are presented on the following pages*

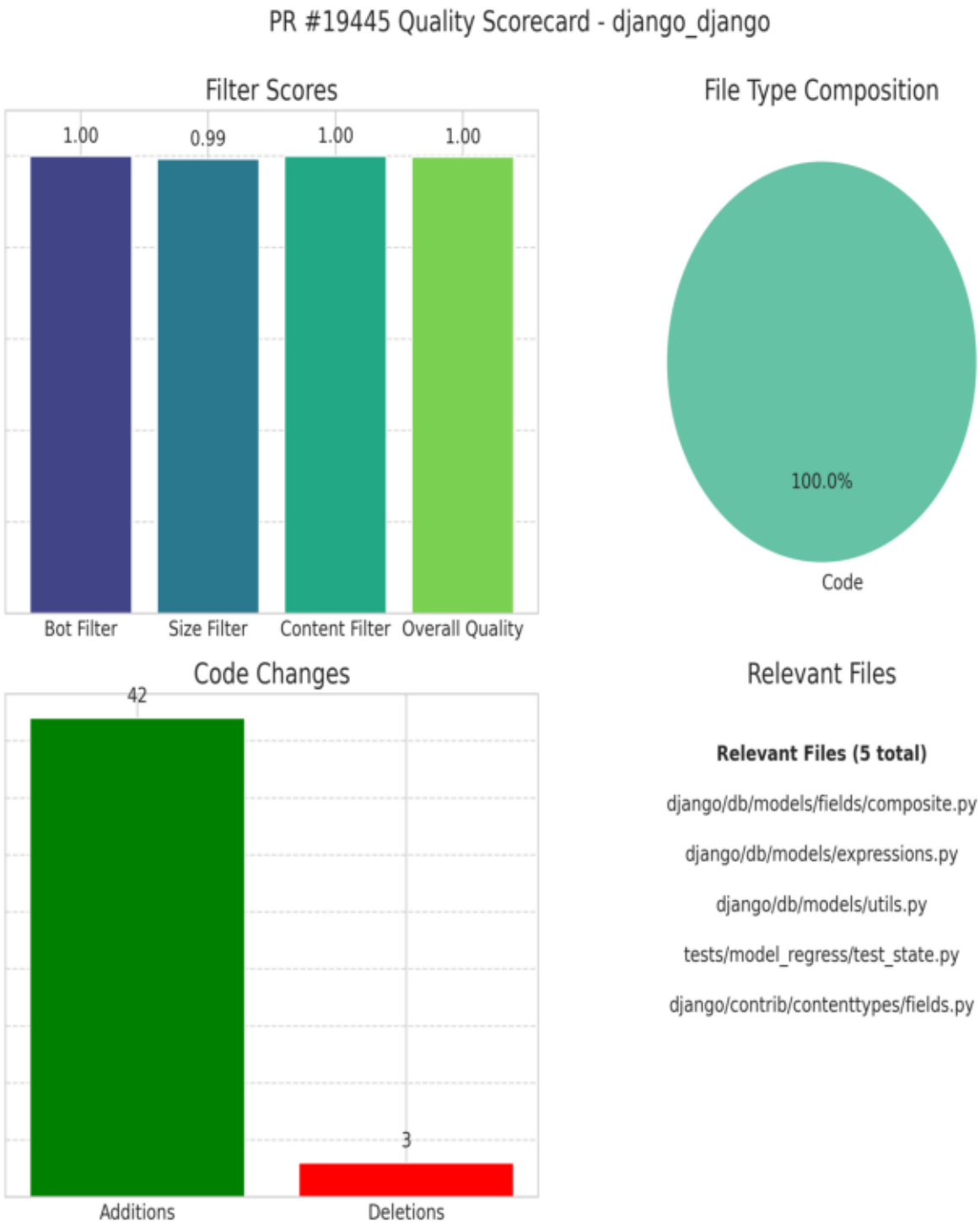
# PR Quality Scorecard: django/django/pr19469



# PR Quality Scorecard: django/django/pr19419



# PR Quality Scorecard: django/django/pr19445



# Methodology

The data curation pipeline implements a multi-stage filtering approach inspired by the SWE-RL paper, focusing on extracting high-quality software engineering data from GitHub repositories. The pipeline consists of the following key components:

## 1. Data Acquisition

- GitHub API integration for PR events and metadata
- Repository cloning for file content access
- Linked issue resolution and context gathering

## 2. Multi-Stage Filtering

- Bot and Automation Detection: Identifies and filters out automated PRs
- Size and Complexity Filtering: Ensures PRs are neither trivial nor unwieldy
- Content Relevance Filtering: Focuses on meaningful software engineering content

## 3. Relevant Files Prediction

- Identifies semantically related files not modified in the PR
- Uses import analysis and directory structure heuristics
- Enhances context for understanding code changes

## 4. Quality Metrics Generation

- Comprehensive quality scoring across multiple dimensions
- Metadata extraction for filtering decisions
- Relevance scoring based on problem-solving indicators

*The filtering pipeline maintains high precision by using progressive refinement, ensuring that only PRs with genuine software engineering value are retained while capturing detailed metadata about filtering decisions.*