# Predict StackOverflow Underrated Answers

Hanhan Wu

# Introduction

## Why StackOverflow?

- Most popular Software Development Community
- Rapidly Growing
- Crowd Sourcing
- Benefits
    - Share knowledge and experience with fellow developers
    - Gain reputation
    - Learning
    - Other

# Motivation

**Studies are showing:**

- StackOverflow is a major channel for developers to get help
- Top Voted or Accepted solution may not be the best solution

**Invoked Questions:**

- Is there any Underrated Answer?
- Is it easy to find Underrated Answers?
- Is there anyway to detect Underrated Answers, automatically?

# Terminology

- **Underrated Answer** - Less votes but better than the top rated solution

- **Better Solution Criteria**
  - Simpler
  - Serves for more platform
  - Still works now
  - More Efficient
  - Closer to the question code?
  - More positive sentiment?

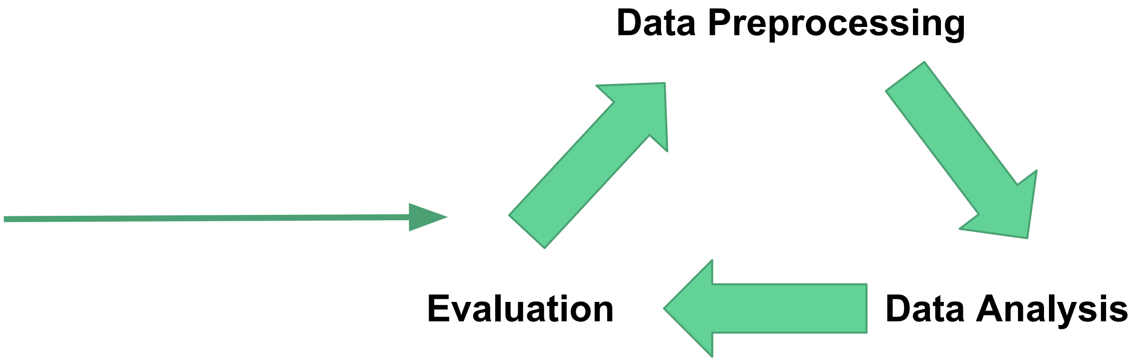
# Goals

- Look into Underrated Answers
  - Are they rare?
  - What makes them better?
  - Why did they get underrated?

- Try to Detect Underrated Answers Automatically
  - Is there any effective method?

# Approach - Overall

1. Data Collection
2. Feature Generation
3. Data Science Workflow
4. Generate Insights



Data Preprocessing

Data Analysis

Evaluation

# Approach - Data Collection

- 107,558 random posts
- “Python” tag, 3+ Answers
- 2000+ Json Files

**Format for Each File:**

- 1 **Post** (id, title, text, votes, favorite count, code)
  - All the **Answers** (id, text, code, vote)
    - All the **Comments** for each Answer (id, text, code, vote)

# Approach - Feature Generation

- **Code Metrics** (46 features)
  - Coding Style, Python Syntax
    - 4 levels - Module, Class, Function, Code
    - E.g. code percentage, comment percentage, class badname, etc.
  - Raw metrics
    - E.g. LOC, LLOC, Multiline Strings, Blank Lines, etc.
  - Cyclomatic Complexity - number of decisions in a code block
    - E.g. complexity of each function, all functions complexity, etc.
  - Analysis through AST tree
    - E.g. number of distinct operators, bugs, difficulty, etc.
  - Maintainability Index

# Approach - Feature Generation

- **Sentiment Analysis** (12 features)
  - Sentence based analysis
  - Comments Sentiment for Each Answer
    - Take Vote Count into consideration
  - Answer Sentiment
  - Format:
    - Sentiment Score
    - Very Positive Count
    - Positive Count
    - Neutral Count
    - Negative Count
    - Very Negative Count

# Approach - Feature Generation

- **Other** (3)
  - Answer Code vs Question Code
    - Sequence Match Score
    - Ignore Junk Items
  - Each Answer vs Top Rated Answer
    - Vote/TotalVote
    - MaxVote - Vote

- **Label -** IsUnderrated
- **IDs**
  - Question ID
  - Answer ID

**Start With 61 Numerical Features**

**Each Row:** QuestionID - AnswerID - Features

# Approach - Data Science Workflow

- Clustering
  - Explore whether there are **grouped patterns**
  - Explore whether Underrated Answers could be grouped together
- Classification
  - Prediction with Ground Truth
  - Explore whether there is an effective prediction method

# Challenges

- Data Collection
  - StackOverflow API cannot link data together
  - Hidden data
  - Text Data Cleaning is troublesome
- Feature Generation
  - Python Version Conflicts & Syntax Error
  - Limited open source output
  - Sentient analysis for text data
- Data Labeling for Classification
  - Crowd Sourcing can be biased, participants need training
  - Manually labeling is also time consuming
- Small Amount of Data

# Current Progress

1. Data Collection (Done)
2. Feature Generation (Done)
3. Data Science Workflow (TO-DO)
4. Generate Insights (TO-DO)

# In case PPT changed format

In Case Link