AutoPyDep: A Recommendation System for Python Dependency Management Utilizing Graph-Based Analytics

Dibyendu Brinto Bose Travis Chan Mathhew Trimble Dr. Chris Brown

Department of Computer Science, Virginia Tech

Introduction and Motivation

Managing Dependencies: As software projects grow, managing dependencies becomes challenging. Conflicting or outdated libraries lead to **"Dependency Hell"**, causing build failures and runtime errors.

Limitations of pip: Tools like **pip** are widely used but struggle with complex dependency chains and version conflicts, making them inadequate for large-scale projects.

Need for Better Solutions: Dependency management is inherently complex (**NP-complete**), requiring smarter tools to address these challenges and streamline development.

Research Questions(RQ)

- **RQ1:** How accurately can the interdependencies between package versions be identified and modeled based on historical release records?
- **RQ2:** How do developers evaluate the usability of *AutoPyDep*?
- **RQ3:** Which features of *AutoPyDep* do developers find most valuable?

Contributions

- **Comprehensive Python Release Notes Dataset**: Includes detailed information on release dates, versions, and categories.
- **TF-IDF+NE-Based Graph Network Transformation**: Transforms release notes into a graph network using TF-IDF and Named Entity (NE) extraction to capture relationships and dependencies between library versions.
- **AutoPyDep Tool**: Offers intuitive features like dependency analysis, relationship mapping, and predictions for release categories/dates.

AutoPyDep features

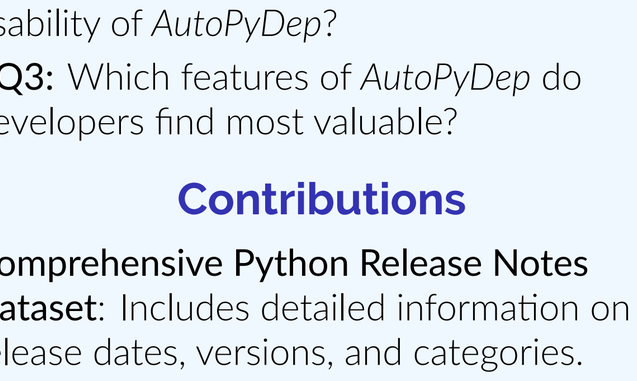
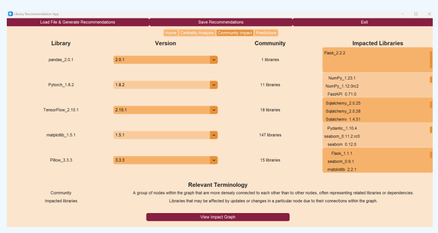
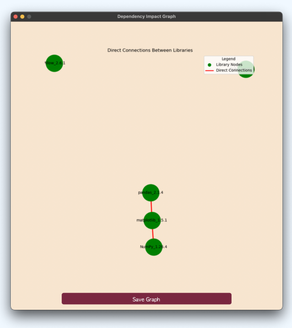


Figure 1. Centrality Analysis tab of AutoPyDep



(a) Interdependencies information



(b) Impact Graph

Figure 2. Community Impact tab of AutoPyDep

Workflow

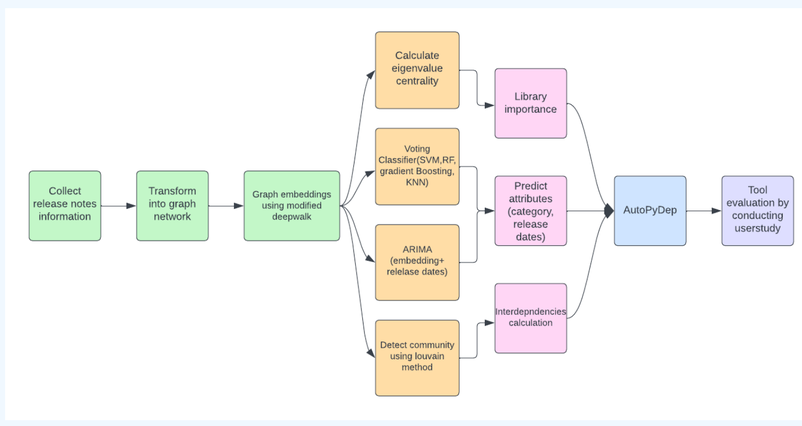


Figure 3. Workflow of our work

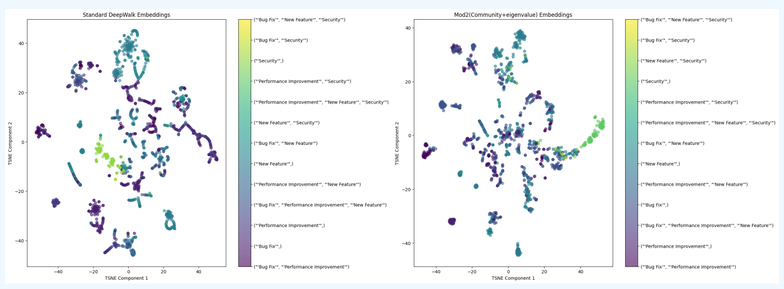
Graph Creation from Release Notes

Input: Release notes dataset D
Output: Graph $G = (V, E)$

1. **Initialize Vertices and Edges:**
   - $V \leftarrow$ Nodes representing each library version in D
   - $E \leftarrow \emptyset$
2. **Feature Extraction:**
   - Compute TF-IDF and apply Named Entity Recognition (NER) to each release note in D
   - Apply Truncated SVD on TF-IDF+NE feature set to reduce dimensions
3. **Edge Creation Based on Similarity:**
   - For each pair of release notes (r_i, r_j):
     - Compute cosine similarity $sim(r_i, r_j)$
     - If $sim(r_i, r_j) > 0.4$, add edge (r_i, r_j) to E
4. **Edge Creation for Consecutive Versions:**
   - For each library L in D:
     - For consecutive versions (v_k, v_{k+1}), add edge (v_k, v_{k+1}) to E
5. **Return Graph:** $G = (V, E)$

Modified DeepWalk Approach

Traditional DeepWalk methods generate numerous random walks, which often result in scattered clusters. Our approach, called *Mod2 (Community + Eigenvalue)*, addresses these limitations by:



(a) Embeddings of Standard DeepWalk

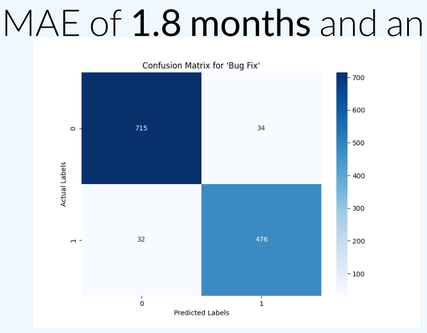
(b) Embeddings of Modified DeepWalk

Figure 4. TSNE Visualization of embeddings

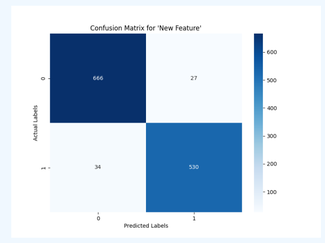
- **Leveraging Eigenvector Centrality**: We identify the most influential nodes in the graph to prioritize key areas of the network.
- **Incorporating Community Detection**: By focusing on densely connected regions, we ensure that random walks traverse critical nodes that capture important relationships between different components.

Answer to RQs

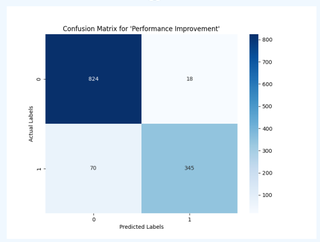
- **Prediction Models:** The category prediction model achieved a median cross-validated F1 score of **0.8**. The date prediction model achieved an MAE of **1.8 months** and an RMSE of **2 months**.



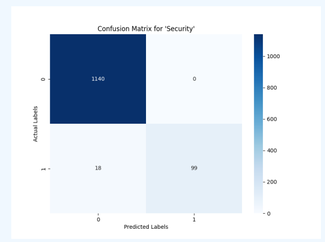
Bug Fix



New Feature



Performance Improvement



Security

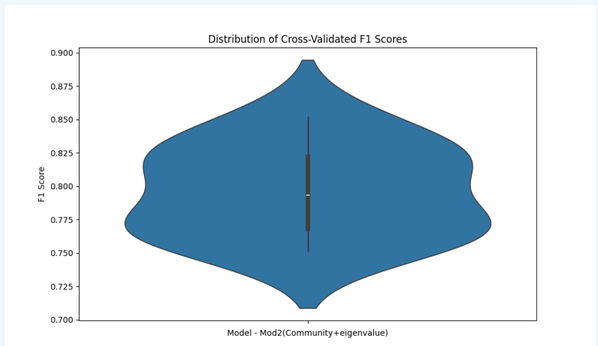


Figure 5. Cross-validation result distribution

- **Usability (RQ2):** The tool received a mean SUS score of **70.54**, consistently rated as **"Good"**.
- **Feature Ratings (RQ3): Direct Connection Graph (4.77)** and **Community Impact (4.55)** were the highest-rated features, while **Next Release Date Prediction (3.36)** was rated the lowest.

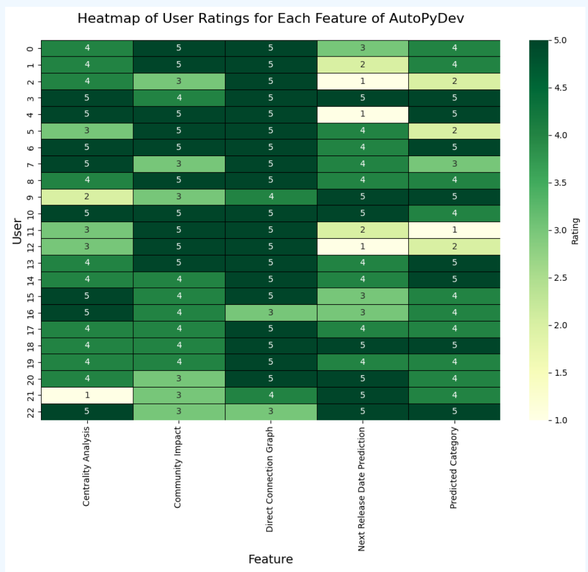


Figure 6. AutoPyDep Feature Ratings

Conclusion

AutoPyDep provides an effective recommendation system for managing Python library dependencies with strong prediction models and favorable user feedback, enhancing dependency management for developers.

Amazon - Virginia Tech Initiative for Efficient and Robust Machine Learning

brintodibyendu@vt.edu