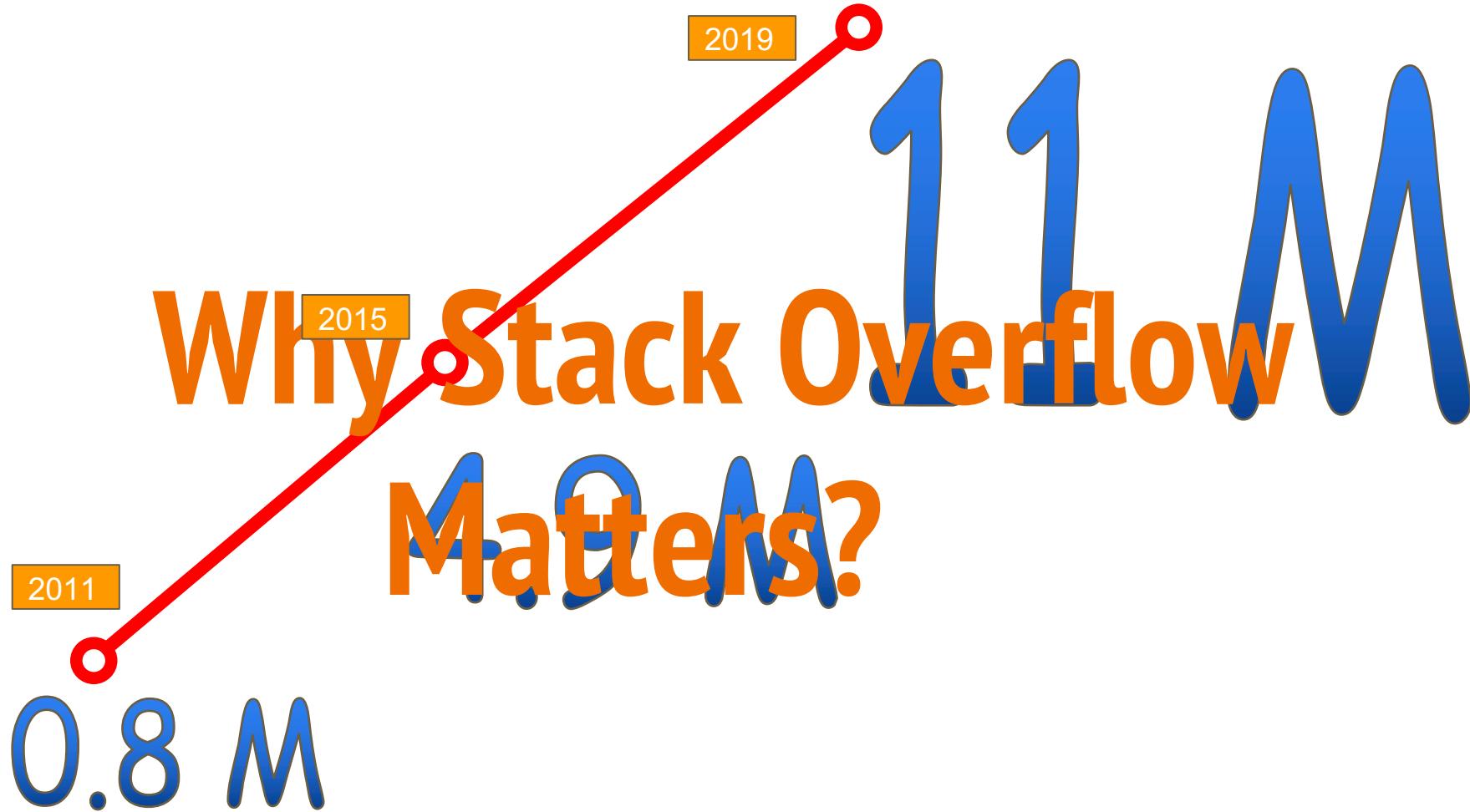


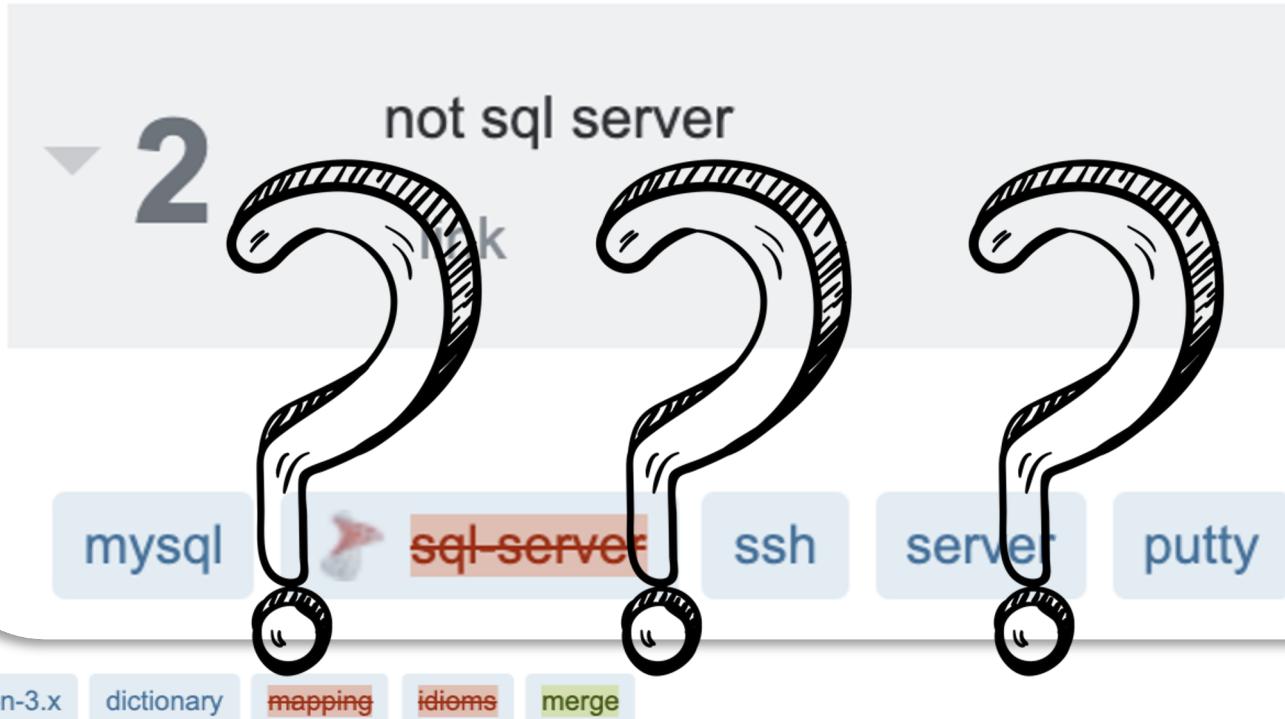
Can stackoverflow Be Better?

Weijia Li, Haoran Chen, Nan Xu

Master's Program in Big Data
Computer Science, Simon Fraser University



Issue One: Tag trustworthy?



Issue Two: Polite enough?

The screenshot shows a post on Stack Overflow. The post has 14 upvotes and no answers. A comment is highlighted with a red box around the text "WTF??!!".

WTF??!! the OP had it right, then an hour later it was BROKEN
upvotes and favs?!?? (btw: lines 705+ in source code)

[source](#) [link](#)

The Goal of this project

- Can we help user select the most accurate tags?
- Can we create a more polite community?
- Can we uncover any hidden relation between technical skills and side interests?

What are we fighting against?

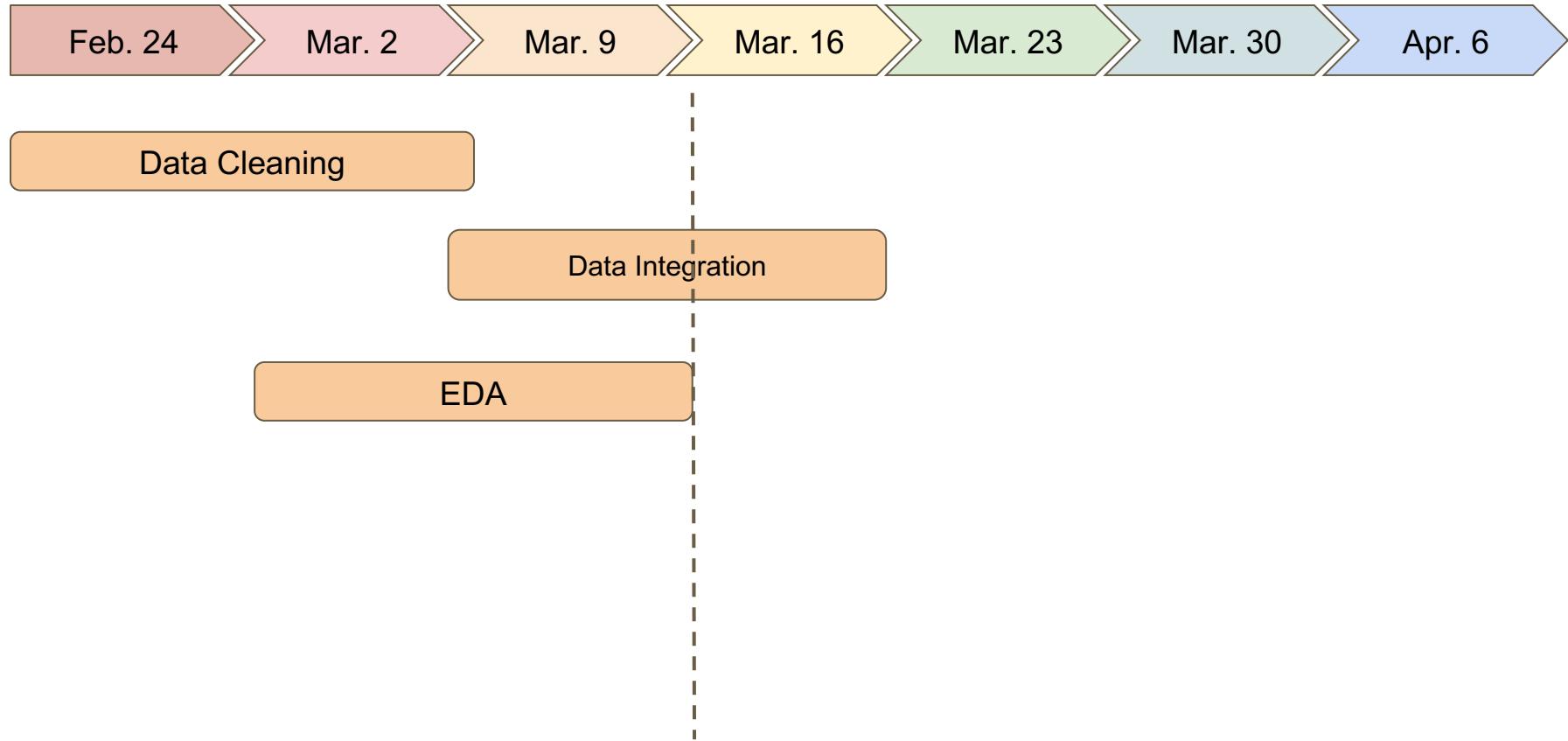
- Large dataset
- Complex data manipulation structure
- Strong knowledge of NLP, CNN, RNN



text
public
processed
understanding
automatic
linguistics
input
tag
typo
discourse
job
analys
word
conference
programming
technology
automated
intelligent
cloud
data
evolution
science
media
summarization
networks
machine
media
connect
artificial
intelligence
design
simulation
communicate
telecommunications
output
operating
topography
information
systems

learning
processing
NLP
natural
language
interaction

What have we done so far?



Source Code

```
def posts_analysys():
    df_posts = pd.read_csv('posts_for_eda_cleaned.csv')
    num_rows = len(df_posts.index)
    num_tags_all = df_posts['TagCounts'].sum()

    df_year = df_posts.groupby(['Year']).mean().drop(['Id'], axis=1)
    print(df_year)

    print('Number of posts by the end of 2017: ' + str(num_rows))
    print('Number of tags of all posts by the end of 2017: ' + str(
    print('Average number of tags of all posts by the end of 2017: '))

def history_analysis():
    df_posts = pd.read_csv('posts_for_eda_cleaned.csv').rename(columns={'PostId': 'PostId'})
    df_posthistory = pd.read_csv('posthistory_for_eda.csv').drop(['PostId'])

    num_rows = len(df_posthistory.index)

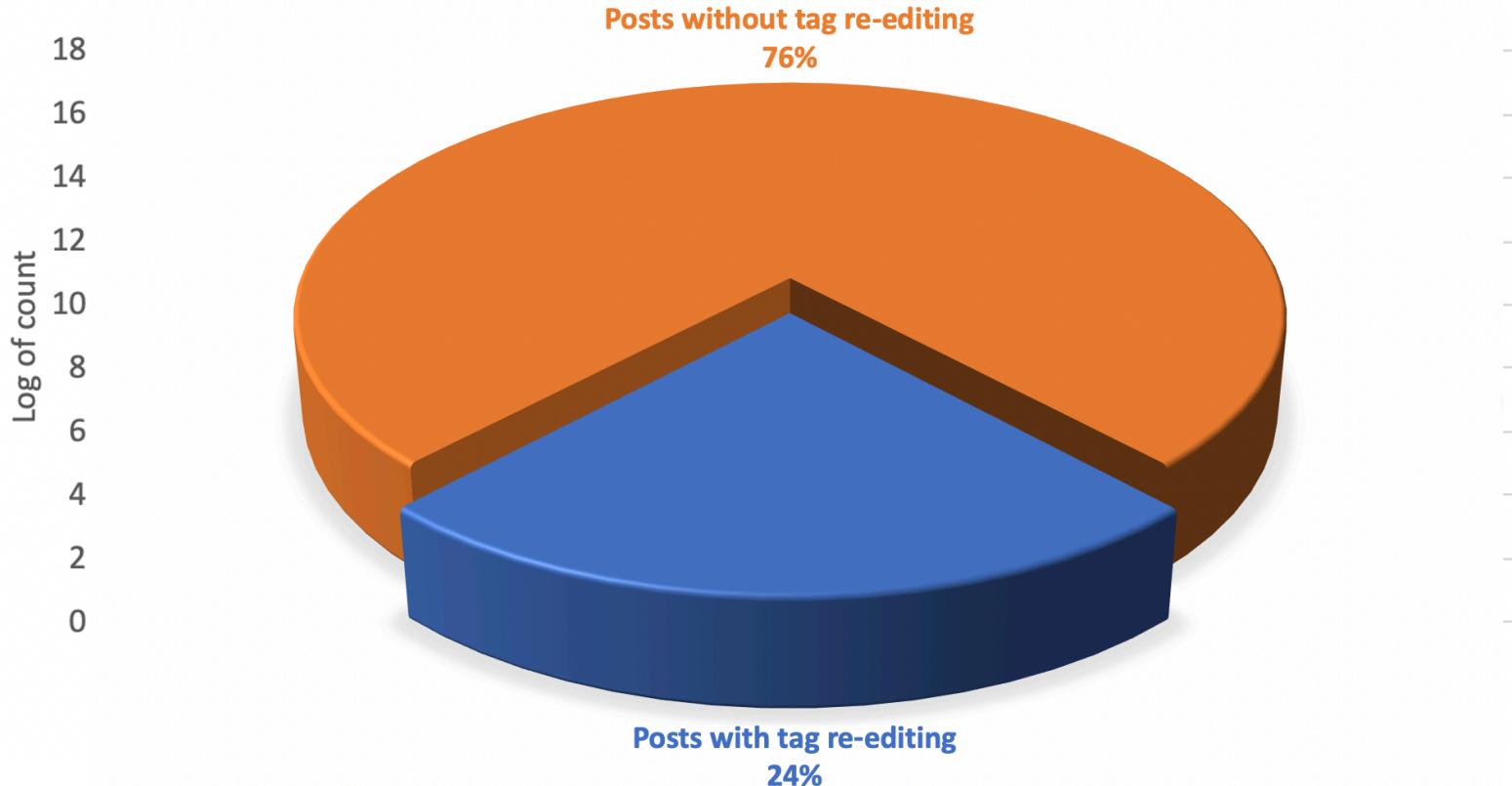
    df_count = df_posthistory.groupby(['PostId']).size().reset_index()

    df_posthistory_year = pd.merge(df_posts, df_count, on='PostId', how='left')

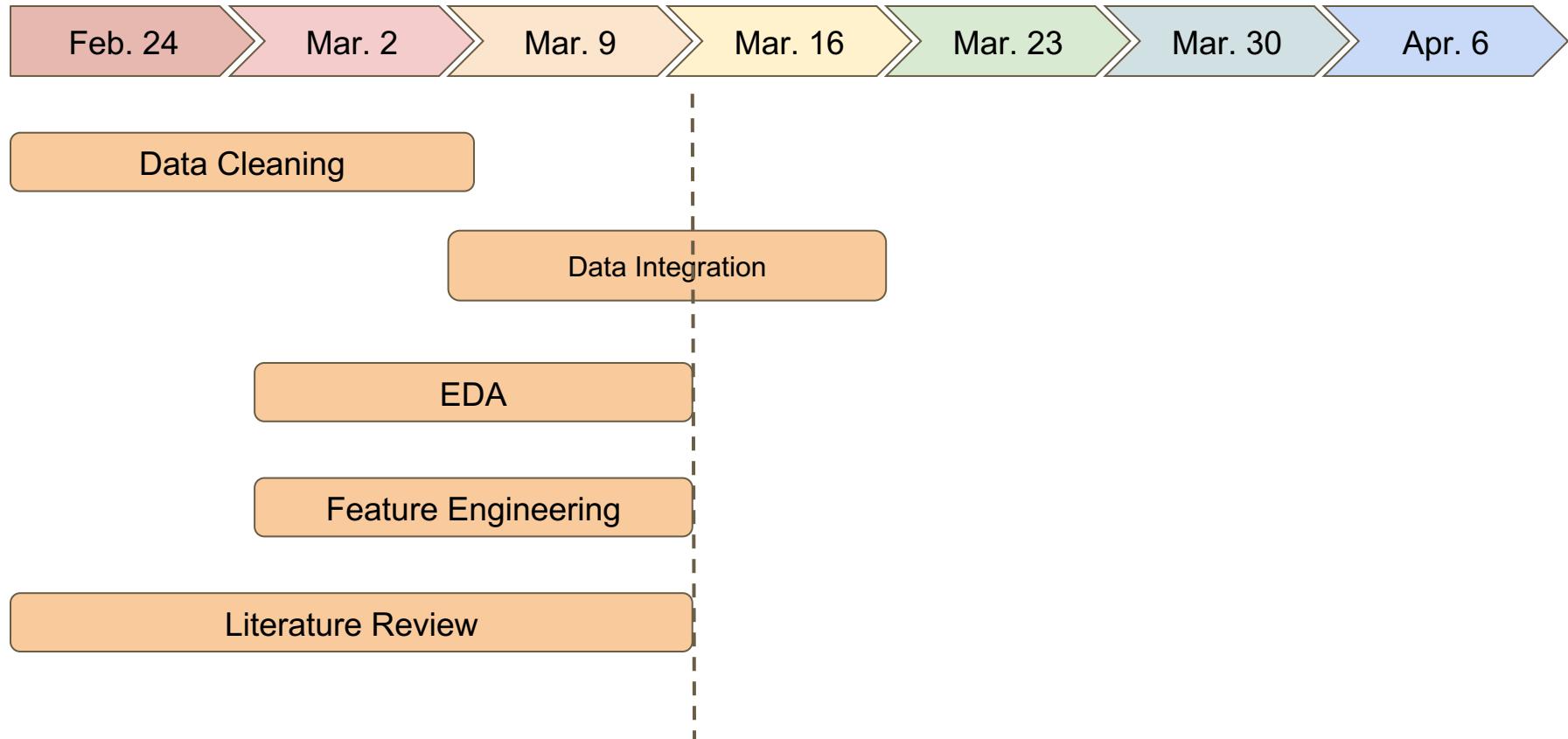
    print('Number of changes in tags: ' + str(num_rows))
    print(df_count.max())
    print(df_posthistory_year)

    df_change = df_count.groupby(['Number of Update']).size()
    print(df_change)
```

EDA Results

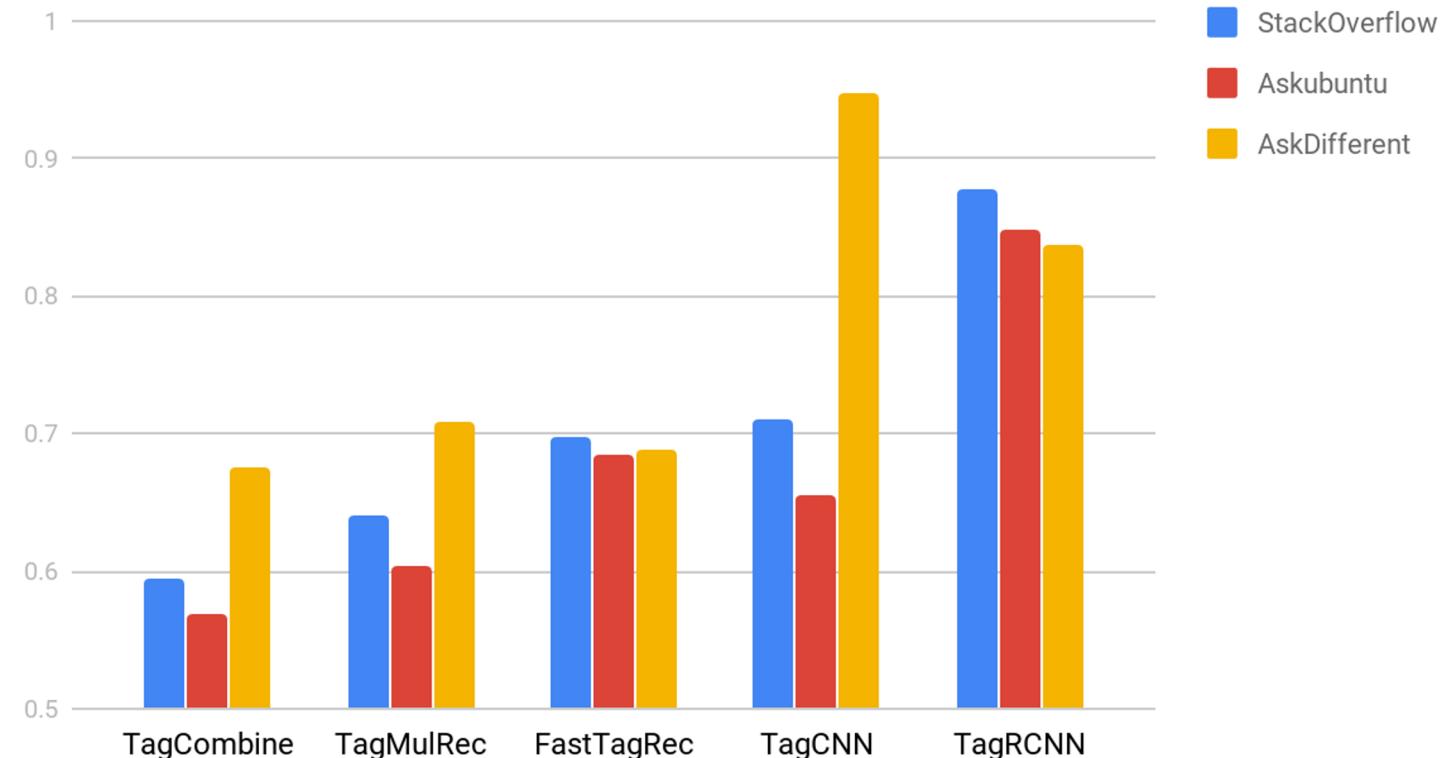


What have we done so far?

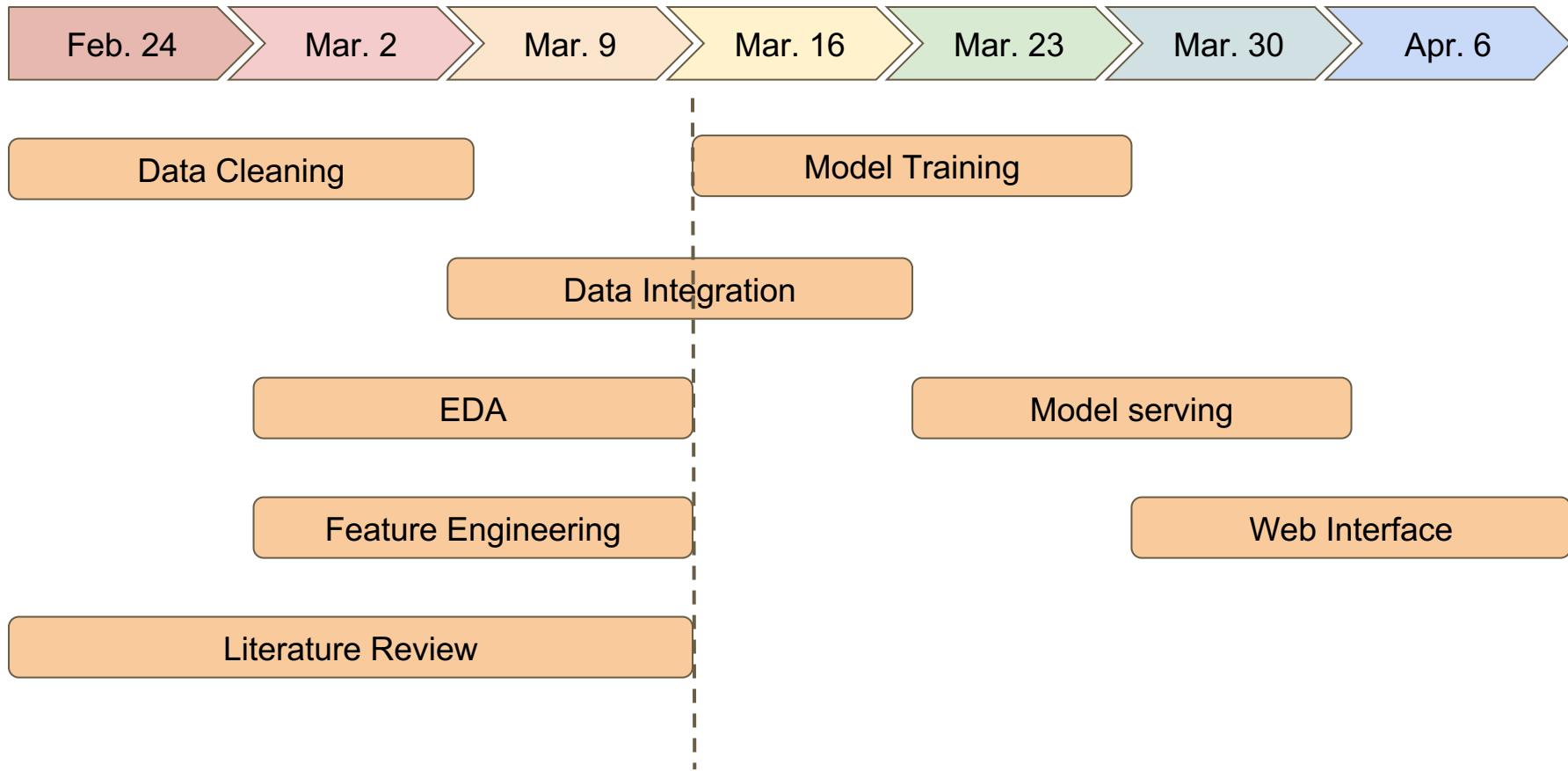


Evolving Models

Recall@5 Across Multiple Platforms



What's next?



Web Interface Prototype

Title

Does Python have a ternary conditional operator?

Body

If Python does not have a ternary conditional operator, is it possible to simulate one using other language constructs?

Tags

python

operators

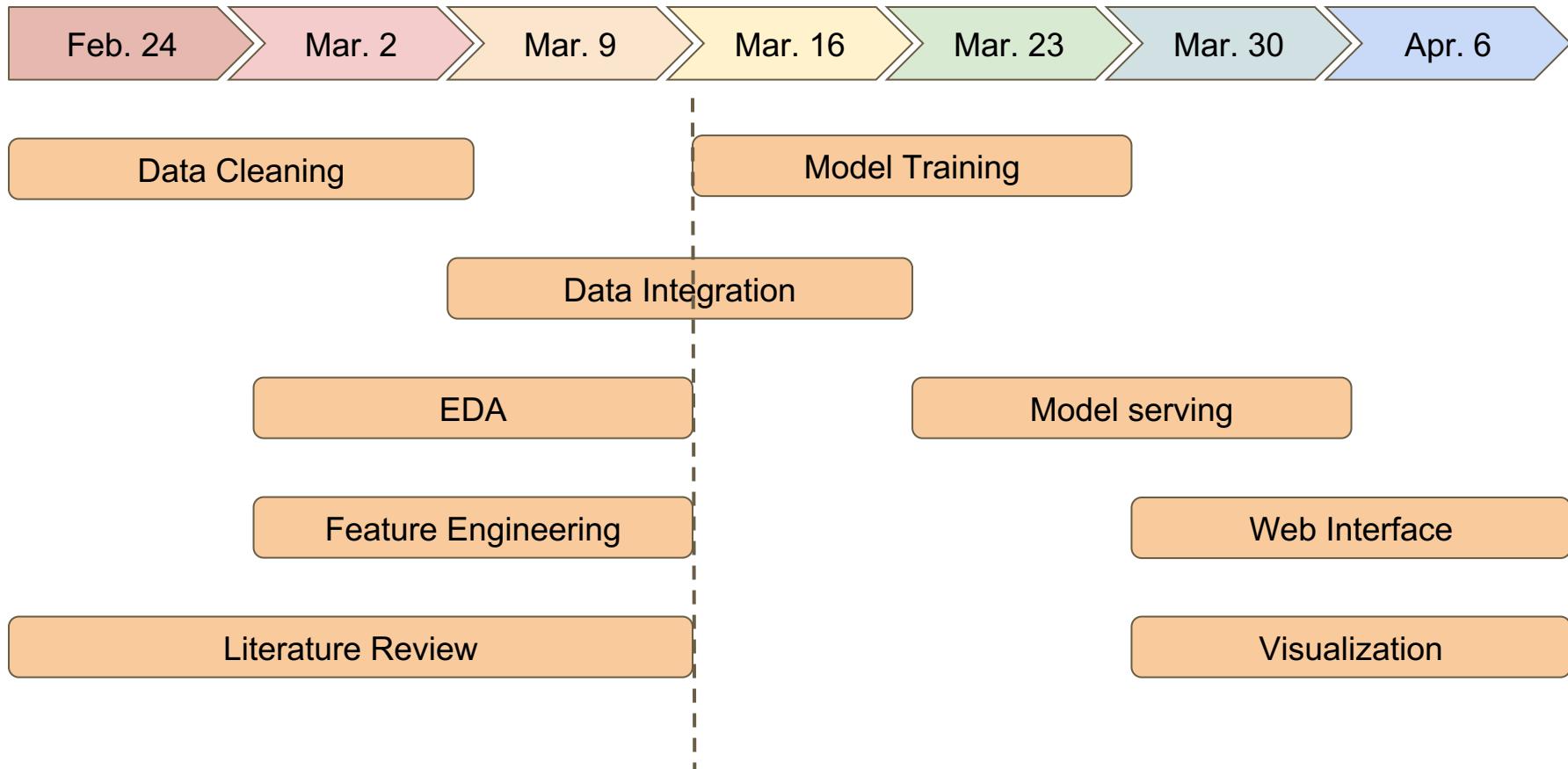
ternary-operator

conditional-operator

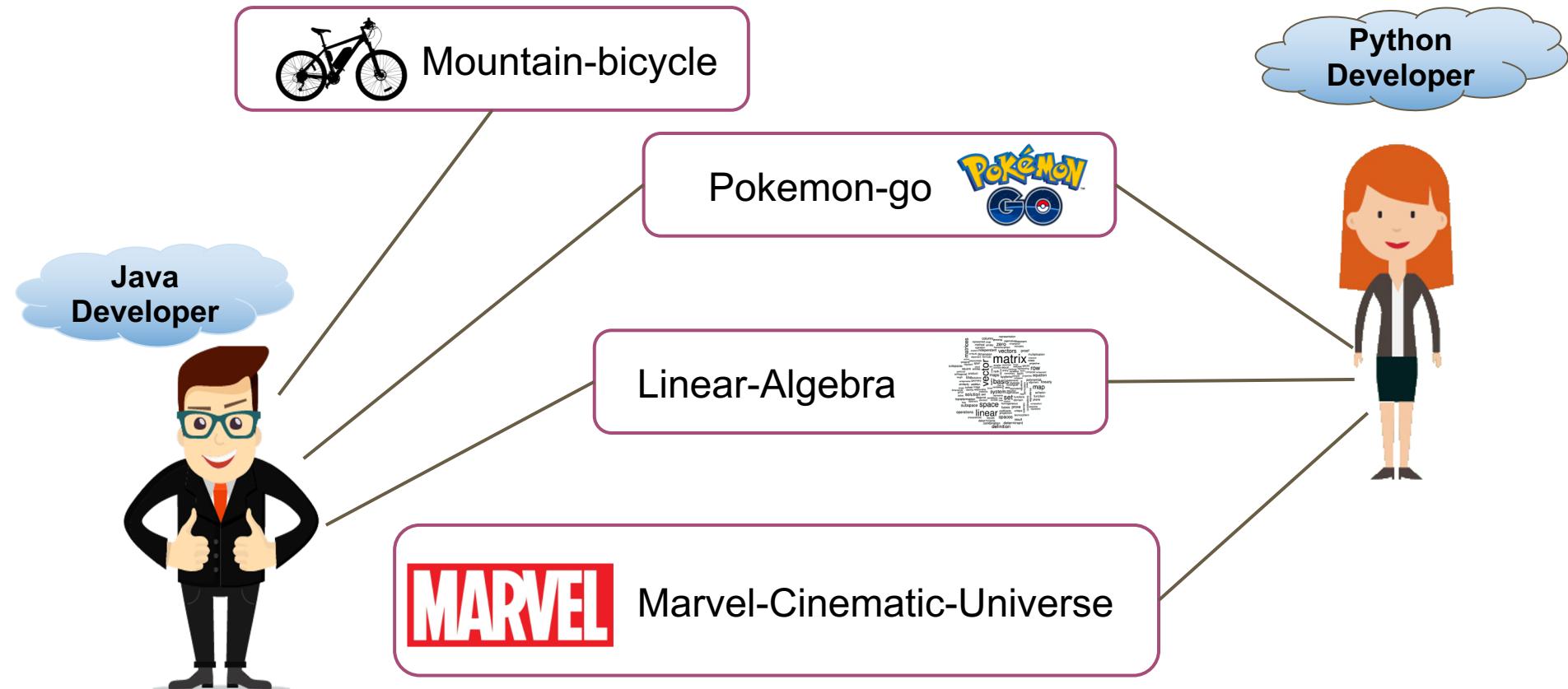
Suggest Tags

Post Question

What's next?



Cross-Platform Potential Relation Analysis



Mitigate Risks

- High individual workload
- Lack of cross-platform users
- Unexpected Technical Difficulty