Analyzing GitHub Comments

in Python and Java Projects

Josef Gugglberger, Till Volkmer

Problem

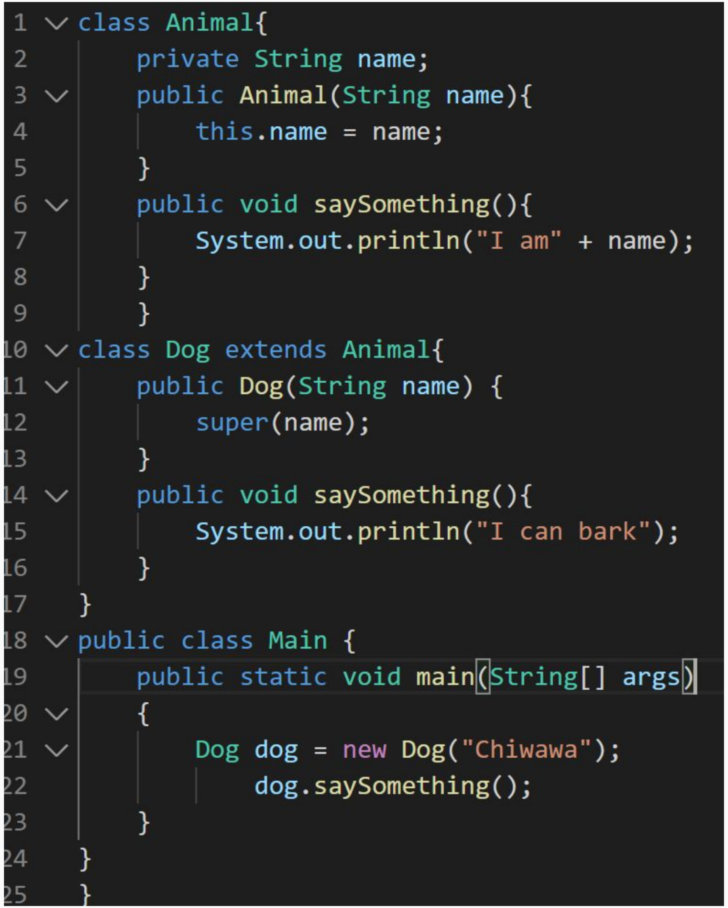
- **Motivation**
    - Comments are important to understand and maintain source code
    - Goal was to compare commenting styles and quantities in Java and Python projects by analyzing comments in Java and Python projects on GitHub

- **Method**
    - formulate research questions
    - Choose 10 popular Open-Source projects on GitHub for each language
    - Extract data from repositories
    - Analyze and visualize data

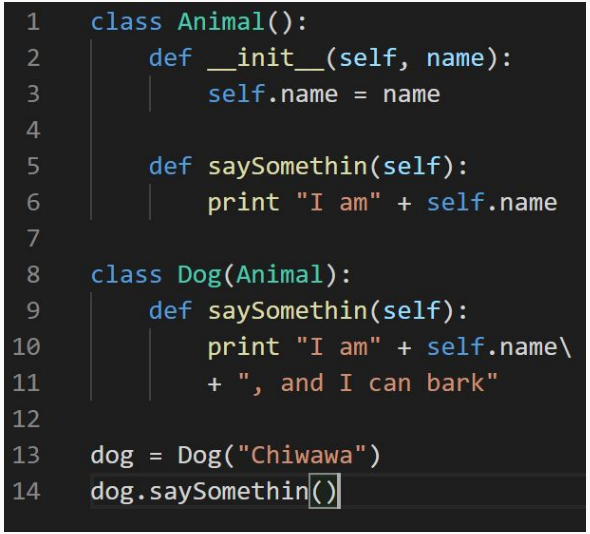
Research questions

- **RQ1**: Is there a correlation between the quantity of comments in source code and the language used?
  - metric: lines of code / lines of comment
  - H_0: Python projects require less comments within the source code
- **RQ2**: Is there a correlation between the popularity on GitHub and the amount of comments?
  - metric: correlation between comments and stars on GitHub
  - H_0: Projects which have are more popular and have a larger amount of contributors have more comments within the source code
- **RQ3**: Is there a correlation between the sentiment in comments in source code and the language used?
  - metric: sentiment score from -1 (negative) to 1 (positive)
  - H_0: Java is more often used by corporate developers, thus the sentiment is more strict

Differences between Java and Python



```java
class Animal{
    private String name;
    public Animal(String name){
        this.name = name;
    }
    public void saySomething(){
        System.out.println("I am" + name);
    }
    }
class Dog extends Animal{
    public Dog(String name) {
        super(name);
    }
    public void saySomething(){
        System.out.println("I can bark");
    }
}
public class Main {
    public static void main(String[] args)
    {
        Dog dog = new Dog("Chiwawa");
            dog.saySomething();
    }
}
}
```



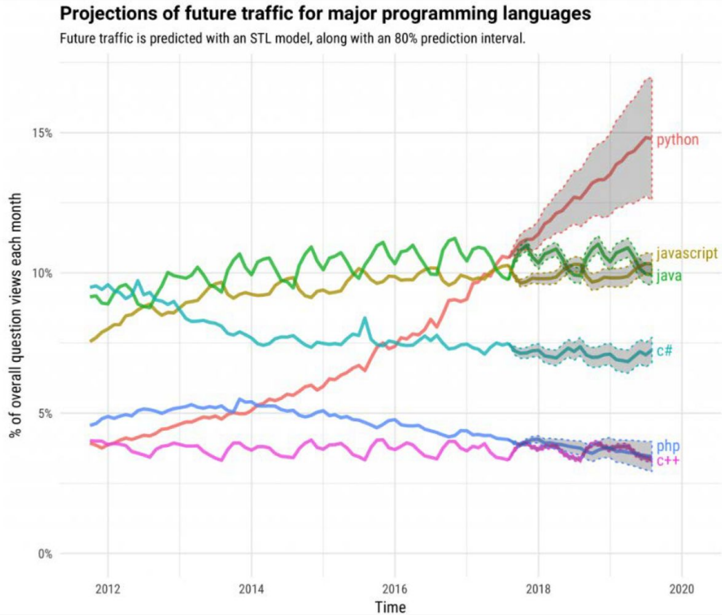
```python
class Animal():
    def __init__(self, name):
        self.name = name

    def saySomethin(self):
        print "I am" + self.name

class Dog(Animal):
    def saySomethin(self):
        print "I am" + self.name\
        + ", and I can bark"

dog = Dog("Chiwawa")
dog.saySomethin()
```

- ➢ Python requires fewer lines of code to reach the same output
- ➢ Java delivers more information within the blank code (definition of each variable)

Differences between Java and Python

Java

- Initial release: 1995
- **compiled** language
- **object oriented** programming language
- **statically** typed

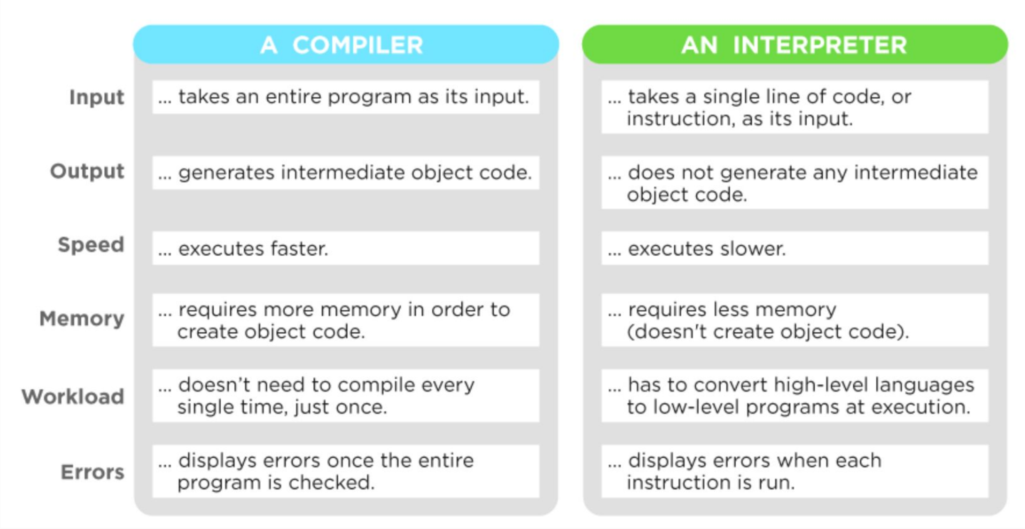


Projections of future traffic for major programming languages

Future traffic is predicted with an STL model, along with an 80% prediction interval.

Python

- Initial release: 1991
- **interpreted** language
- **scripting language**
- **dynamically** typed
- is considered to be **easier to read/understand**



	A COMPILER	AN INTERPRETER
Input	... takes an entire program as its input.	... takes a single line of code, or instruction, as its input.
Output	... generates intermediate object code.	... does not generate any intermediate object code.
Speed	... executes faster.	... executes slower.
Memory	... requires more memory in order to create object code.	... requires less memory (doesn't create object code).
Workload	... doesn't need to compile every single time, just once.	... has to convert high-level languages to low-level programs at execution.
Errors	... displays errors once the entire program is checked.	... displays errors when each instruction is run.

Analyzed Projects

- Java
  - spring-boot
  - fastjson
  - guava
  - jmeter
  - RxJava
  - mockito
  - dubbo
  - zxing
  - elasticsearch
  - okhttp

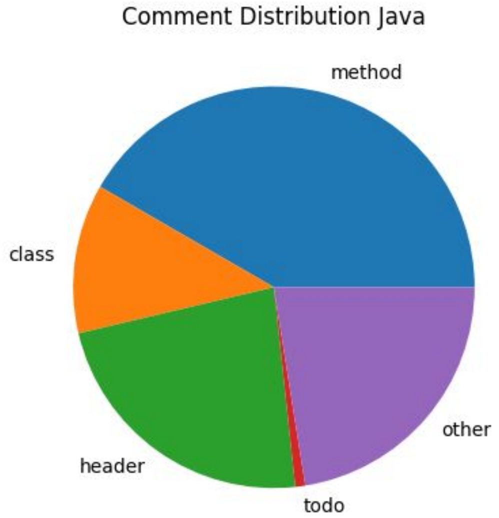
total lines (code and comments) = **4.109.897**

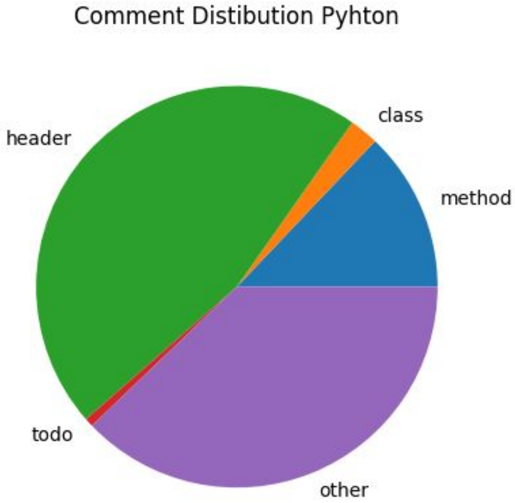
- Python
  - ansible
  - airflow
  - keras
  - spaCy
  - scikit-learn
  - pandas
  - tornado
  - scrappy
  - flask
  - django

total lines = **3.090.293**

Comment categorization

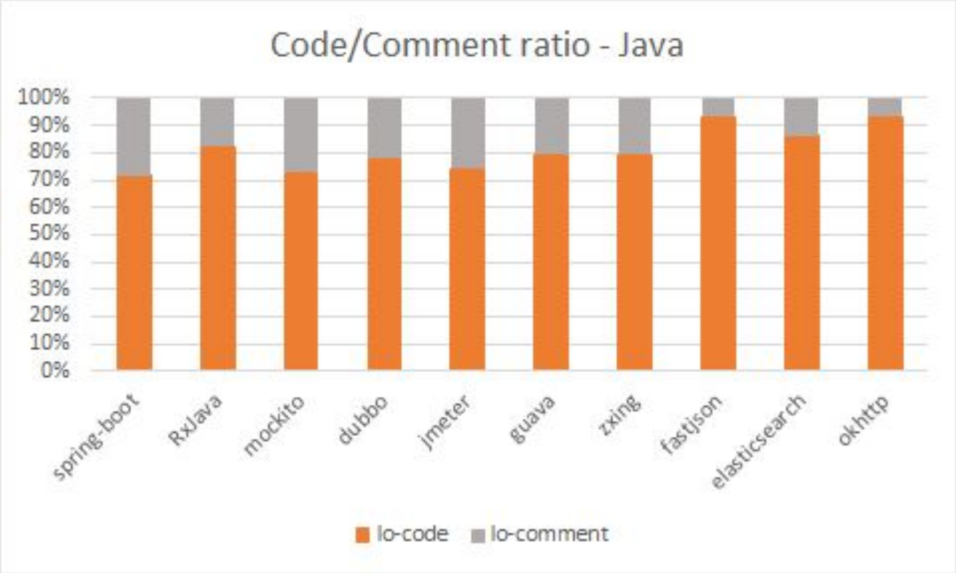
1. Header comments
   - overview information or copyright information
2. Method comments
   - describe the functionality of a method
3. Inline comments
   - describe implementation decisions within a method body
4. Class comments
   - describe the functionality of a class
5. Task comments
   - notes containing a remaining todo or a remark

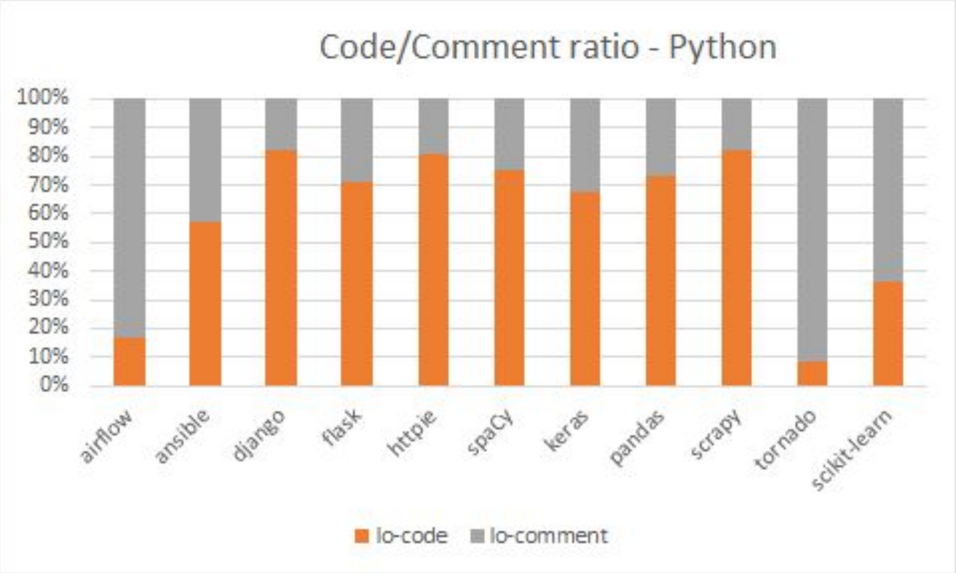




RQ1 Is there a correlation between the quantity of comments in source code and the language used?

- H_0: Python projects require less comments within the source code
  - ○ avg(lo comment/code Python) < avg (lo comment/code Java)
- H_a: There are more contributing factors which affect comment quantities
  - ○ avg(lo comment/code Python) >= avg (lo comment/code Java)





➢ Python projects have more lines of comments on average, consequently a higher comment to code ratio

Lines of comment/ lo code

Python:	Java:
4,86384409	0,395414765
0,759325763	0,214637274
0,220857416	0,371506628
0,403268318	0,284776048
0,240344295	0,348144444
0,323261531	0,257888075
0,486062477	0,262321981
0,365129932	0,073328566
0,219515902	0,160617452
11,13819514	0,074234894

avg(Python) = 1,901980486

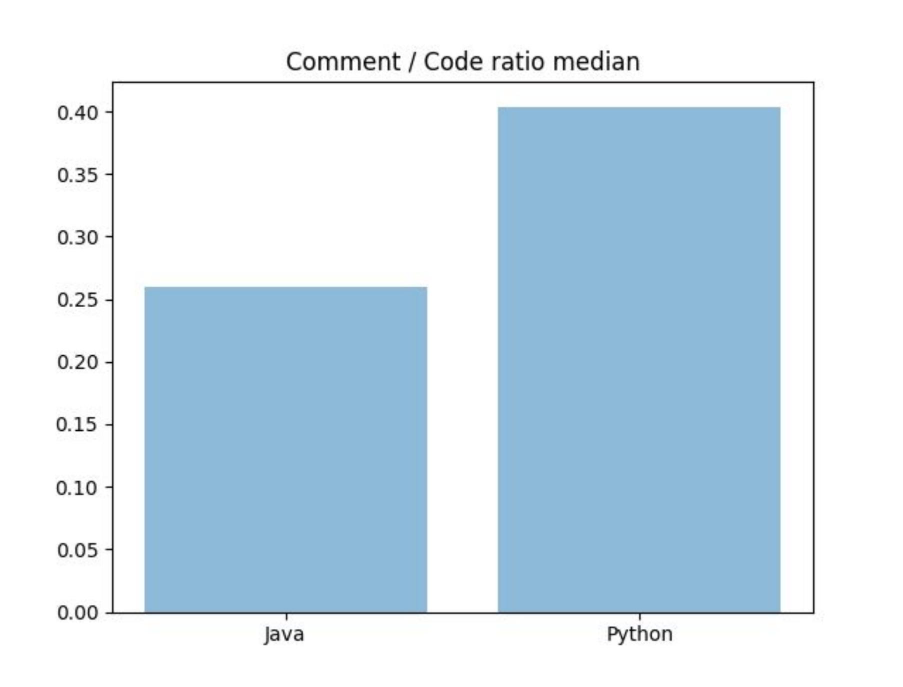
Without outliers: avg(Python) = **37,7%**

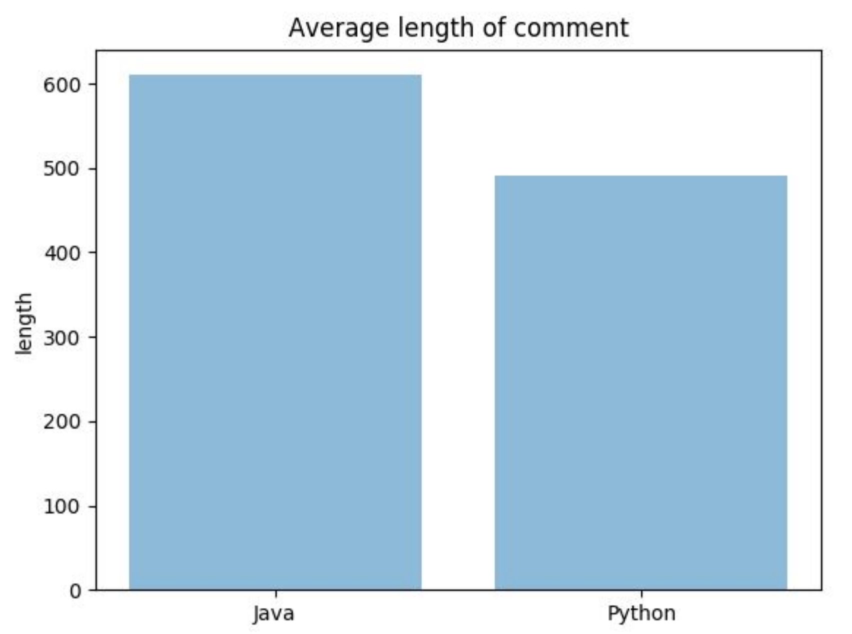
avg(Java) = 0,244287012

Without outliers: avg(Java) = **29,6%**

- H_0: Python projects require less comments within the source code
  - avg(Python) = **0,3772** < avg(Java) = **0,2961** **rejected**
- H_a: There are more contributing factors which affect comment quantities
  - avg(lo comment/code Python) >= avg (lo comment/code Java)

RQ1 Is there a correlation between the quantity of comments in source code and the language used?



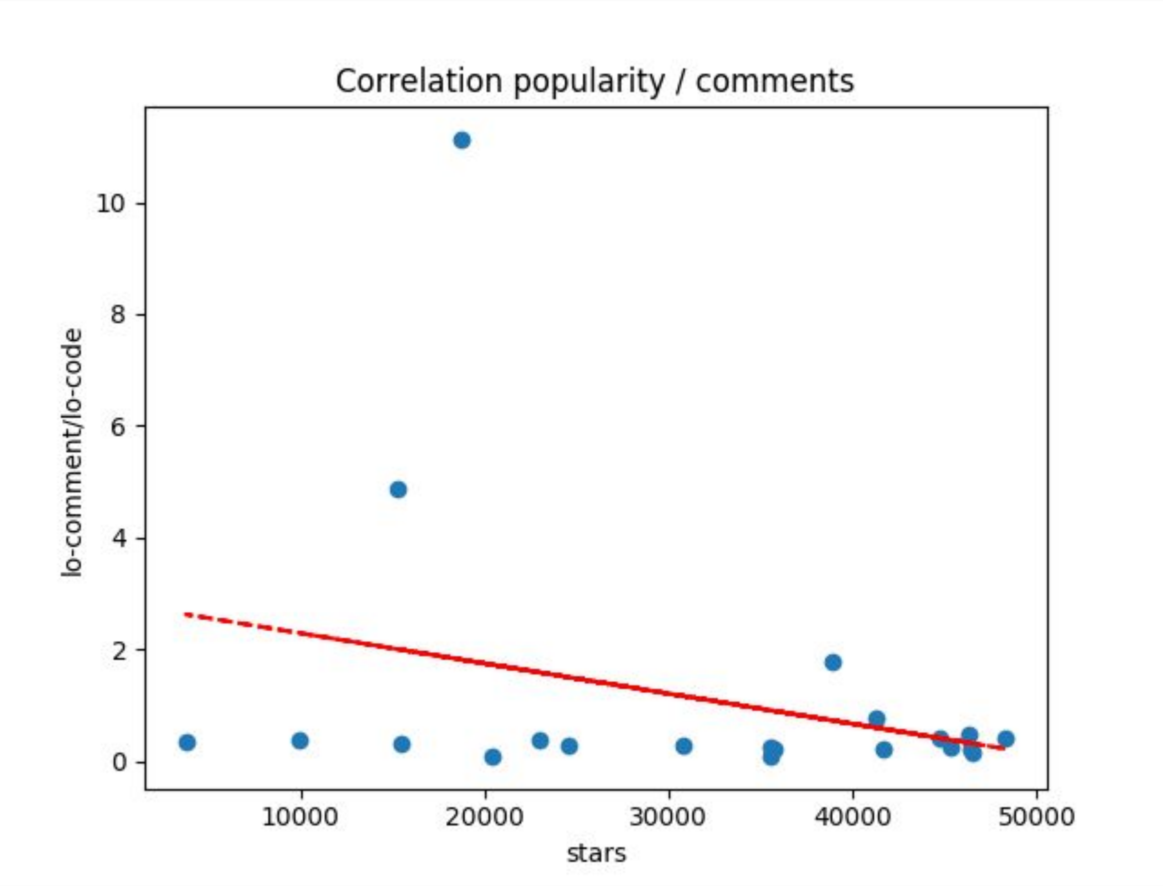


- **Despite the analyzed Python projects have a higher ratio of comments within the source code, the average length per comment is larger in the analyzed Java projects**

RQ2

Is there a correlation between the popularity on GitHub and the amount of comments?

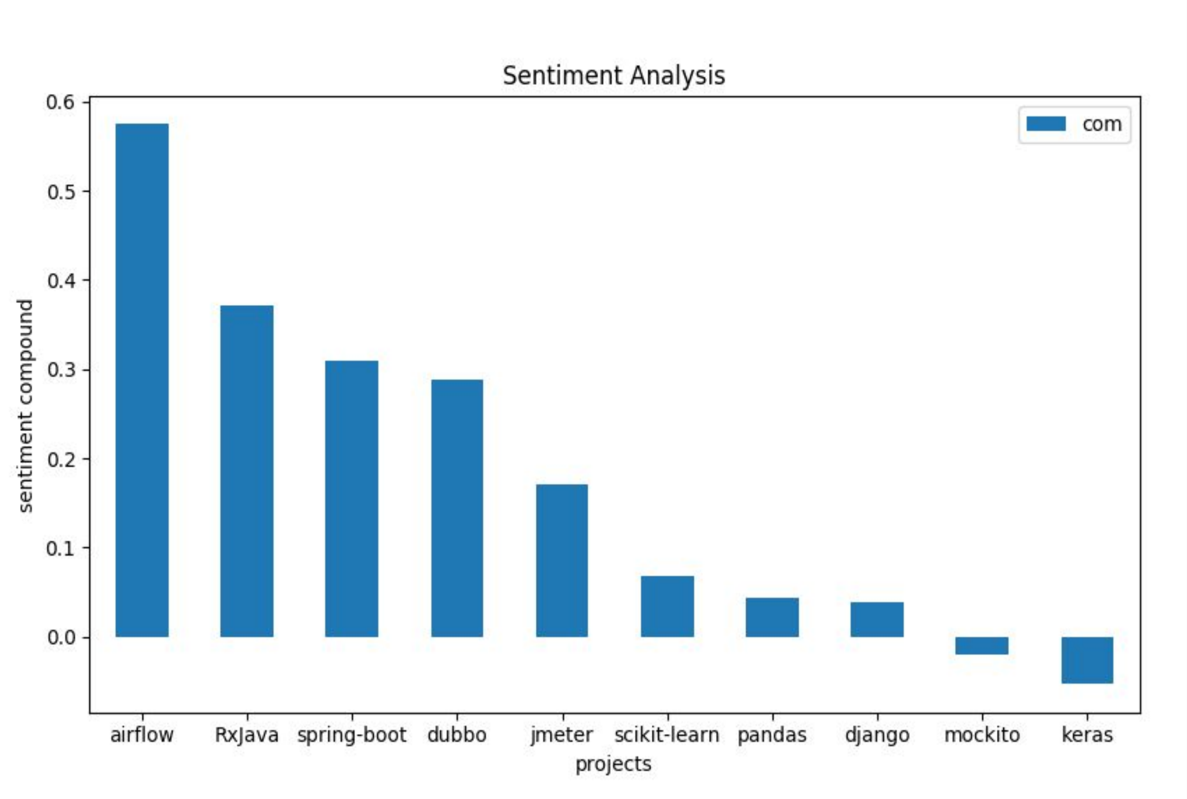
- popularity seems to have an impact on number of comments
- but to less data points



RQ3

Is there a correlation between the sentiment in comments in source code and the language used?

- Python package VADER was used for sentiment analysis
- Text in comments is rather in an analytical and describing style
- We could **not** acknowledge the results of VADER.



Conclusion

Findings:

- Python code is more extensively documented than Java Code
- No correlation between popularity and amount of comments
- Sentiment analysis does not make much sense in source code comments

TODO:

- Analyse additional factors: Domain, Lifetime, Maintainer