GitHub Code Review Smells

In [1]:
```
import pandas as pd
import numpy as np
from datetime import datetime
from datetime import timedelta
from collections import Counter
import ast
import os
import re
import json
from preprocess import github_utils
```

Generating CSV tables

In [2]:
```
project_names = ['desktop','vscode','tensorflow','django']
for project in project_names: github_utils.generate_table(project)
```

```
Generated the table for project: desktop
Generated the table for project: vscode
Generated the table for project: tensorflow
Generated the table for project: django
```

Preprocess-Reading Data:

In [3]:
```
def size_labelling(changed_loc):
    if changed_loc==0:
        size='0'
    elif 1<=changed_loc<10:
        size = 'XS'
    elif 10<=changed_loc<50:
        size = 'S'
    elif 50<=changed_loc<200:
        size = 'M'
    elif 200<=changed_loc<1000:
        size = 'L'
    elif 1000<=changed_loc:
        size = 'XL'
    return size
```

In [4]:
```
def read_data(project):
    generic = lambda x: ast.literal_eval(x)
    conv = {'reviewers': generic, 'dismissed': generic,
            'comments': generic, 'reviews': generic}
    df = pd.read_csv(f'tables/{project}.csv', converters=conv, error_bad_lines=True,parse_dates=True) # can be deleted.
    df['created_at'] = pd.to_datetime(df['created_at'])
    df['merged_at'] = pd.to_datetime(df['merged_at'])
    df.drop(df[ df.changed_loc == 0 ].index, inplace=True)
    df.drop(df[ df.is_merged != True ].index, inplace=True)
    df.drop(df[df.author == ('Deleted user', 'ghost')].index, inplace=True)
    df['size_label'] = df.apply(lambda row: size_labelling( row.changed_loc ), axis=1)
    return df
```

In [5]:
```
projects = [read_data(project) for project in project_names]
```

```
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\1078632602.py:5: FutureWarning: The error_bad_lines argument has been
deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'tables/{project}.csv', converters=conv, error_bad_lines=True,parse_dates=True) # can be deleted.
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\1078632602.py:5: FutureWarning: The error_bad_lines argument has been
deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'tables/{project}.csv', converters=conv, error_bad_lines=True,parse_dates=True) # can be deleted.
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\1078632602.py:5: FutureWarning: The error_bad_lines argument has been
deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'tables/{project}.csv', converters=conv, error_bad_lines=True,parse_dates=True) # can be deleted.
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\1078632602.py:5: FutureWarning: The error_bad_lines argument has been
deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'tables/{project}.csv', converters=conv, error_bad_lines=True,parse_dates=True) # can be deleted.
```

In [6]:
```
tot = 0
for name,df in zip(project_names,projects):
    tot += len(df)
    print(f'Project {name} has {len(df)} instances.')

print(tot)
```

```
Project desktop has 2993 instances.
Project vscode has 5206 instances.
Project tensorflow has 9807 instances.
Project django has 5578 instances.
23584
```

Smell #1 : Lack of Code Review:

In [7]:
```
def self_review_check(author,reviewers):
    return reviewers == [author]

def no_review_check(reviewers):
    return reviewers == []

def detect_lack_of_reviews(review):
    return self_review_check(review.author, review.reviewers) or no_review_check(review.reviewers)
```

In [8]:
```
all_smelling = 0
all_instances = 0
for name,df in zip(project_names,projects):
    df['lack_of_review'] = df.apply(lambda review:
                             detect_lack_of_reviews(review),axis=1
                            )
    #print(name + '==>', end=' ')
    all_smelling += len(df[df['lack_of_review']==True])
    all_instances += len(df)
    s = "{:<15} ==> {} smelling instances over {}. Smell percentage (%): {:.2f}".format(name, len(df[df['lack_of_review']=
    #print(str(len(df[df['lack_of_review']==True])) + ' smelling instances over ' + str(len(df)) + '.  Smell percentage(%)
    print(s)
print('total' + '==>', end=' ')
print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '.  Smell percentage(%): ' + str(round(100* a
```

```
desktop         ==> 440 smelling instances over 2993. Smell percentage (%): 14.70
vscode          ==> 3000 smelling instances over 5206. Smell percentage (%): 57.63
tensorflow      ==> 1276 smelling instances over 9807. Smell percentage (%): 13.01
django          ==> 3341 smelling instances over 5578. Smell percentage (%): 59.90
total==> 8057 smelling instances over 23584.  Smell percentage(%): 34.2
```

Smell #2 : Review Buddies

In [9]:
```
def detect_review_buddies(df, min_commit_size = 50):
    """
    This function returns the ratio between the developers
    having a review buddy and not having a review buddy.
    The developers that have been chosen have more commits
    than the given min_commit_size parameter.
    """

    df = df[df.lack_of_review==False]
    merged = df.groupby('author').reviewers.apply(list).to_dict()
    #print("total devs:" + str(len(merged)))
    count_gt_buddy = 0
    count_gt = 0

    for (author, reviews) in merged.items():
        freq = np.array(list(Counter([reviewer
                                      for reviewers in reviews
                                      for reviewer in reviewers
                                      if reviewer != author]).values()))

        if (len(reviews) >= min_commit_size) and len(freq)>0:
            count_gt += 1
            freq = freq/freq.sum()
            if (freq.max() >= 0.5):
                count_gt_buddy += 1

    return (count_gt, count_gt_buddy)
```

In [10]:
```
all_buddies = 0
all_devs = 0

for name, df in zip(project_names,projects):
    print(name + '==>', end=' ')
    devs, devs_with_buddy = detect_review_buddies(df)
    all_buddies += devs_with_buddy
    all_devs += devs
    print(str(devs_with_buddy) + ' devs with a review buddy over ' + str(devs) + '.  Smell percentage(%): ' + str(round(10

print('total' + '==>', end=' ')
print(str(all_buddies) + ' devs with a review buddy over ' + str(all_devs) + '.  Smell percentage(%): '  + str(round(100*a
```

```
desktop==> 0 devs with a review buddy over 6.  Smell percentage(%): 0.0
vscode==> 1 devs with a review buddy over 13.  Smell percentage(%): 7.7
tensorflow==> 1 devs with a review buddy over 31.  Smell percentage(%): 3.2
django==> 1 devs with a review buddy over 9.  Smell percentage(%): 11.1
total==> 3 devs with a review buddy over 59.  Smell percentage(%): 5.1
```

Smell # 3: Ping-pong:

In [11]:
```
def read_commits(project):

    df = pd.read_csv(f'commits_freeze/{project}.csv', error_bad_lines=True)
    df['commit_data'] = df.apply(lambda x:
                                 json.loads(r'{}'.format(x.commit_data)),
                                 axis=1
                                )

    df = df[['id','commit_data']]
    return df
```

In [12]:
```
commit_dfs = [read_commits(project) for project in project_names]
merged_dfs = [project_df.merge(commit_df, on='id', how='inner') for project_df,commit_df in zip(projects,commit_dfs)]

del commit_dfs
del projects
```

```
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\2378510041.py:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'commits_freeze/{project}.csv', error_bad_lines=True)
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\2378510041.py:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'commits_freeze/{project}.csv', error_bad_lines=True)
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\2378510041.py:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'commits_freeze/{project}.csv', error_bad_lines=True)
C:\Users\kbaci\AppData\Local\Temp\ipykernel_11124\2378510041.py:3: FutureWarning: The error_bad_lines argument has been deprecated and will be removed in a future version. Use on_bad_lines in the future.


  df = pd.read_csv(f'commits_freeze/{project}.csv', error_bad_lines=True)
```

In [13]:
```

def detect_ping_pong(row, iter_threshold=3):
    state_set = set(['APPROVED', 'CHANGES_REQUESTED', 'COMMENTED'])
    state_list = []

    for commit in row.commit_data:
        state_list.append((pd.to_datetime(commit['commit']['author']['date']),'c'))

    for review in row.reviews:
        if review['state'] in state_set:
            state_list.append((pd.to_datetime(review['submitted_at']),'r'))

    for comment in row.comments:
        if comment['user_data'] and comment['user_data']['login'] != row.author:
            state_list.append((pd.to_datetime(comment['created_at']),'r'))

    state_list.sort()

    iterations = 1 if state_list[0][1]=='r' else 0

    for i in range(len(state_list)-1):
        if state_list[i][1] == 'c' and state_list[i+1][1] == 'r':
            iterations += 1

    return (iterations, iterations > iter_threshold)
```

In [14]:
```
all_smelling = 0
all_instances = 0

for name,df in zip(project_names,merged_dfs):
    df['ping_pong'] = df.apply(lambda row:
                                    detect_ping_pong(row)[1],
                                     axis=1
                                    )
    df['iterations'] = df.apply(lambda row:
                                    detect_ping_pong(row)[0],
                                     axis=1
                                    )

    all_smelling += len(df[df['ping_pong']==True])
    all_instances += len(df)
    print(name + '==>', end=' ')
    print(str(len(df[df['ping_pong']==True])) + ' smelling instances over ' + str(len(df)) + '.  Smell percentage(%): ' +

print('total' + '==>', end=' ')
print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '. Smell percentage(%): ' + str(round(100* al
```

```
desktop==> 209 smelling instances over 2993.  Smell percentage(%): 7.0
vscode==> 92 smelling instances over 5206.  Smell percentage(%): 1.8
tensorflow==> 449 smelling instances over 9807.  Smell percentage(%): 4.6
django==> 7 smelling instances over 5578.  Smell percentage(%): 0.1
total==> 757 smelling instances over 23584. Smell percentage(%): 3.2
```

Smell # 4: Sleeeping Reviews:

In [15]:
```
def detect_sleeping_review(review):
    return (review.merged_at - review.created_at) >= pd.Timedelta('2 days')
```

In [16]:
```
all_smelling = 0
all_instances = 0

for name,df in zip(project_names,merged_dfs):
    df['sleeping_review'] = df.apply(lambda row:
                                    detect_sleeping_review(row),
                                     axis=1
                                    )
    all_smelling += len(df[df['sleeping_review']==True])
    all_instances += len(df)
    print(name + '==>', end=' ')
    print(str(len(df[df['sleeping_review']==True])) + ' smelling instances over ' + str(len(df)) + '.  Smell percentage(%)

print('total' + '==>', end=' ')
print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '. Smell percentage(%): ' + str(round(100* al
```

```
desktop==> 1240 smelling instances over 2993.  Smell percentage(%): 41.4
vscode==> 2089 smelling instances over 5206.  Smell percentage(%): 40.1
tensorflow==> 4690 smelling instances over 9807.  Smell percentage(%): 47.8
django==> 1887 smelling instances over 5578.  Smell percentage(%): 33.8
total==> 9906 smelling instances over 23584. Smell percentage(%): 42.0
```

Smell #5: Missing PR Description:

In [32]:
```
def detect_missing_description(subject, message):
    linked_issue_exists = False
    short_description_exists = False

    if pd.isna(message) or pd.isna(subject):
        short_description_exists = True
    else:
        if len(message.split('\n'))<2:
            short_description_exists = True

        if re.findall(r"#[0-9]+",message) or 'fixes' in message.lower() or 'ticket' in message.lower():
            linked_issue_exists = True
    return (not linked_issue_exists) and short_description_exists
```

In [33]:
```
all_smelling = 0
all_instances = 0

for name,df in zip(project_names,merged_dfs):
    df['missing_description'] = df.apply(lambda row:
                                         detect_missing_description(str(row.subject), str(row.message)),
                                         axis=1
                                        )
    all_smelling += len(df[df['missing_description']])
    all_instances += len(df)
    print(name + '==>', end=' ')
    print(str(len(df[df['missing_description']==True])) + ' smelling instances over ' + str(len(df)) + '. Smell percentage

print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '. Smell percentage(%): ' + str(round(100* al
```

```
desktop==> 335 smelling instances over 2993. Smell percentage(%): 11.2
vscode==> 1277 smelling instances over 5206. Smell percentage(%): 24.5
tensorflow==> 4330 smelling instances over 9807. Smell percentage(%): 44.2
django==> 2138 smelling instances over 5578. Smell percentage(%): 38.3
8080 smelling instances over 23584. Smell percentage(%): 34.3
```

Smell 7: Large Changesets:

In [19]:
```
def detect_large_changeset(changed_loc):
    """Returns True if the changeset consists of more
    than 500 changed lines of code, False otherwise.

    Keyword arguments:
    changed_loc -- the number of changed lines of code (int).
    """
    return changed_loc >= 500
```

In [20]:
```
all_smelling = 0
all_instances = 0

for name,df in zip(project_names,merged_dfs):
    df['large_changeset'] = df.apply(lambda row:
                                     detect_large_changeset(row.changed_loc),
                                     axis=1
                                    )
    all_smelling += len(df[df['large_changeset']==True])
    all_instances += len(df)
    print(str(len(df[df['large_changeset']==True])) + ' smelling instances over ' + str(len(df)) + '. Smell percentage(%):

print('total' + '==>', end=' ')
print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '. Smell percentage(%): ' + str(round(100* al
```

```
160 smelling instances over 2993. Smell percentage(%): 5.3
415 smelling instances over 5206. Smell percentage(%): 8.0
975 smelling instances over 9807. Smell percentage(%): 9.9
162 smelling instances over 5578. Smell percentage(%): 2.9
total==> 1712 smelling instances over 23584. Smell percentage(%): 7.3
```

Combined Smell Analysis:

In [21]:
```
def combined_smell_check(review):
    """Returns True if the changeset suffers from
    at least one of the code review smells.

    Keyword arguments:
    review -- the code review instance (Pandas dataframe row).
    """
    return review['lack_of_review'] or review['ping_pong'] or review['sleeping_review'] or \
                               review['missing_description'] or review['large_changeset']
```

In [22]:
```
all_smelling = 0
all_instances = 0

for name,df in zip(project_names,merged_dfs):
    df['combined_smell'] = df.apply(lambda row:
                                    combined_smell_check(row),
                                    axis=1
                                   )
    all_smelling += len(df[df['combined_smell']==True])
    all_instances += len(df)
    print(str(len(df[df['combined_smell']==True])) + ' smelling instances over ' + str(len(df)) + '. Smell percentage(%):

print('total' + '==>', end=' ')
print(str(all_smelling) + ' smelling instances over ' + str(all_instances) + '. Smell percentage(%): ' + str(round(100* al
```

```
1868 smelling instances over 2993. Smell percentage(%): 62.4
4316 smelling instances over 5206. Smell percentage(%): 82.9
8015 smelling instances over 9807. Smell percentage(%): 81.7
4990 smelling instances over 5578. Smell percentage(%): 89.5
total==> 19189 smelling instances over 23584. Smell percentage(%): 81.4
```