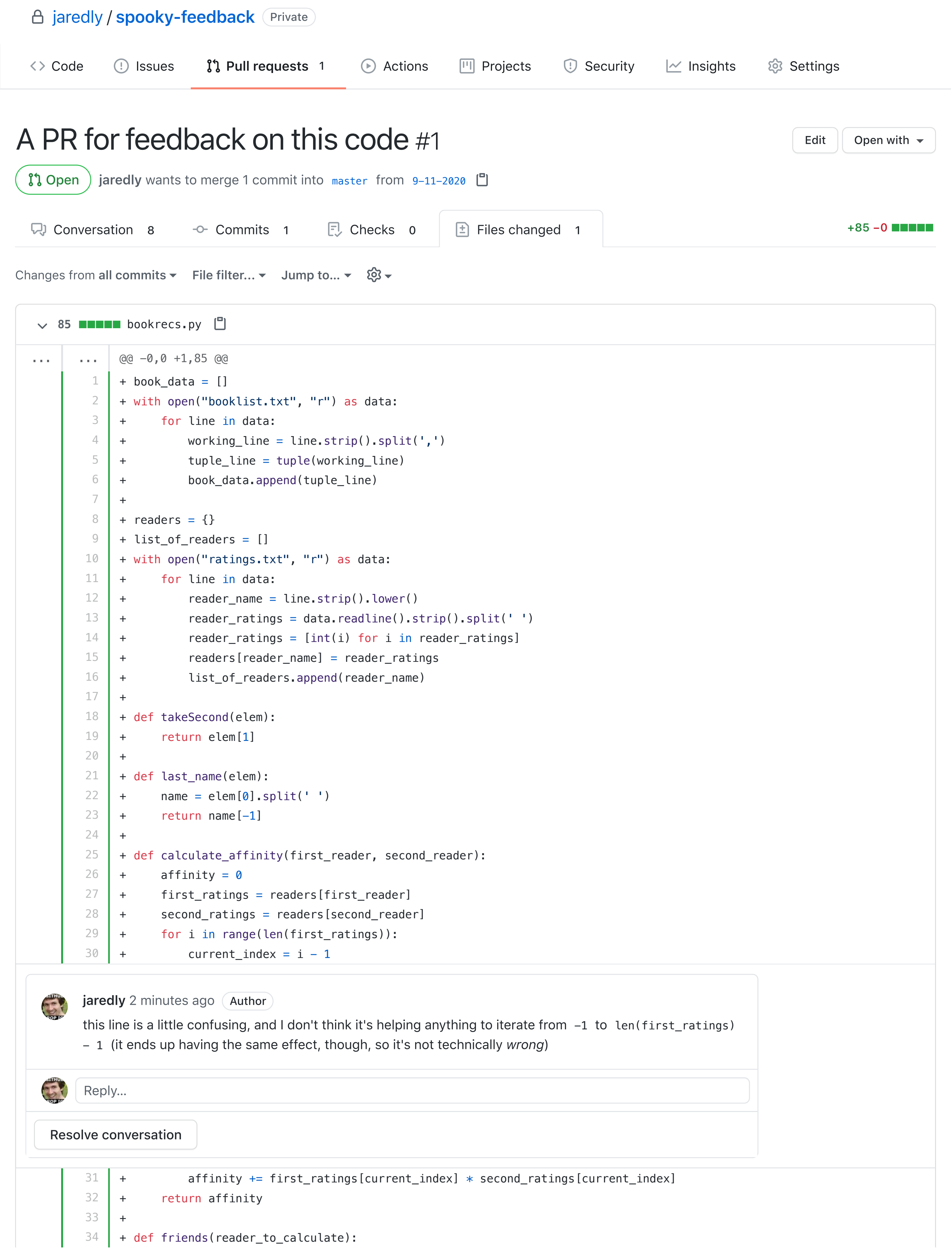

A PR for feedback on this code #1

jaredly wants to merge 1 commit into `master` from `9-11-2020`

```
@@ -0,0 +1,85 @@
+ book_data = []
+ with open("booklist.txt", "r") as data:
+     for line in data:
+         working_line = line.strip().split(',')
+         tuple_line = tuple(working_line)
+         book_data.append(tuple_line)
+
+ readers = {}
+ list_of_readers = []
+ with open("ratings.txt", "r") as data:
+     for line in data:
+         reader_name = line.strip().lower()
+         reader_ratings = data.readline().strip().split(' ')
+         reader_ratings = [int(i) for i in reader_ratings]
+         readers[reader_name] = reader_ratings
+         list_of_readers.append(reader_name)
+
+ def takeSecond(elem):
+     return elem[1]
+
+ def last_name(elem):
+     name = elem[0].split(' ')
+     return name[-1]
+
+ def calculate_affinity(first_reader, second_reader):
+     affinity = 0
+     first_ratings = readers[first_reader]
+     second_ratings = readers[second_reader]
+     for i in range(len(first_ratings)):
+         current_index = i - 1
```

jaredly 2 minutes ago Author

this line is a little confusing, and I don't think it's helping anything to iterate from `-1` to `len(first_ratings) - 1` (it ends up having the same effect, though, so it's not technically *wrong*)

```
+         affinity += first_ratings[current_index] * second_ratings[current_index]
+     return affinity
+
+ def friends(reader_to_calculate):
```

jaredly 2 minutes ago Author

this function could definitely benefit from a docstring, probably something like "finds the two people with the highest affinity score to the given reader"

Reply...

Resolve conversation

```
35  +     scores = []
36  +     for name in readers:
37  +         if reader_to_calculate != name:
38  +             current_affinity = calculate_affinity(reader_to_calculate, name)
39  +             name_score = [name, current_affinity]
40  +             scores.append(name_score)
41  +     scores.sort()
```

jaredly 2 minutes ago Author

this sort can be removed (you're re-sorting the list differently on the next line)

Reply...

Resolve conversation

```
42  +     sorted_friends = sorted(scores, key=takeSecond, reverse = True)
```

jaredly 2 minutes ago Author

another way to do this, that would allow you to use the simpler `.sort()` function, would be to have `name_score` on line `39` have the score first, and then the name. And then this next line would be `(sorted_friends[0][1], sorted_friends[1][1])` as well.
Again this, way isn't wrong, but a little bit more complicated.

Reply...

Resolve conversation

```
43  +     return (sorted_friends[0][0], sorted_friends[1][0])
44  +
45  + def recommend(book_hunter):
46  +     best_friends = friends(book_hunter)
47  +     recommendations = []
48  +     friend1_recommendations = readers[best_friends[0]]
49  +     friend2_recommendations = readers[best_friends[1]]
50  +     friend_likes = []
51  +     hunter_has_read = readers[book_hunter]
52  +     for i in range(len(friend1_recommendations)):
53  +         current_index = i - 1
```

jaredly 2 minutes ago Author

this also looks like it doesn't have a meaningful impact

Reply...

Resolve conversation

```
54  +         book_like_status = [friend1_recommendations[current_index], friend2_recommendations[current_index]]
55  +         friend_likes.append(book_like_status)
56  +     for i in range(len(hunter_has_read)):
```

```
57 +         ind = i - 1
```

jaredly 2 minutes ago Author

same

Reply...

Resolve conversation

```
58 +         current_book = book_data[ind]
59 +         if hunter_has_read[ind] == 0:
60 +             if (friend_likes[ind][0] > 2) or (friend_likes[ind][1] > 2):
61 +                 recommendations.append(current_book)
62 +     recommendations.sort(key=last_name)
63 +     return recommendations
64 +
65 + def recommendations_list(reader_name):
66 +     best_friends = friends(reader_name)
```

jaredly 2 minutes ago Author

you're calculating best_friends twice, doing some unnecessary work -- one thing you could do is have the `recommend` function take `best_friends` as an argument.
so `good_books = recommend(reader_name, best_friends)` , and then not do the calculation on line 46.

Reply...

Resolve conversation

```
67 +     good_books = recommend(reader_name)
68 +     print(f"Recommendtions for {reader_name} from {best_friends[0]} and {best_friends[1]}:")
69 +     for book in good_books:
70 +         print(f"\t {book[0]}, {book[1]}")
71 +
72 +
73 + def main():
74 +     reader_to_test = input("Enter a reader's name: ")
75 +     while reader_to_test != "":
76 +         if reader_to_test in list_of_readers:
77 +             recommendations_list(reader_to_test)
78 +             pass
79 +         else:
80 +             print(f"No such reader {reader_to_test}")
81 +         reader_to_test = input("Enter a reader's name: ")
82 +
83 +
84 + if __name__ == "__main__":
85 +     main()
```

ProTip! Use n and p to navigate between commits in a pull request.