```
from google.colab import drive
drive.mount('/content/drive')

%cd "/content/drive/MyDrive/Annotation project"
```

```
Mounted at /content/drive
/content/drive/MyDrive/Annotation project
```

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split


# Read the adjudicated.txt file into a DataFrame
df = pd.read_csv('adjudicated.txt', sep='\t', header=None)

# Display the DataFrame
print(df)
```

```
        0           1        2  \
0       1  adjudicated     good
1       2  adjudicated     poor
2       3  adjudicated     good
3       4  adjudicated     poor
4       5  adjudicated  average
..    ...          ...      ...
495   496  adjudicated     good
496   497  adjudicated  average
497   498  adjudicated     good
498   499  adjudicated  average
499   500  adjudicated  average

                                                     3
0    This is a special book. It started slow for ab...
1    Recommended by Don Katz. Avail for free in Dec...
2    A fun, fast paced science fiction thriller. I ...
3    Recommended reading to understand what is goin...
4    I really enjoyed this book, and there is a lot...
..                                                 ...
495  4.5 stars! Man of My Dreams is a sweet, emotio...
496  Beat of the Heart is Book Two in the Runaway T...
497  I read Freeing Asia and am a big fan of E.M. A...
498  4 1/2 stars!! Freeing Asia was one of those bo...
499  Sometimes a book comes along that grabs you fr...

[500 rows x 4 columns]
```

```
df.head()
```

	0	1	2	3
0	1	adjudicated	good	This is a special book. It started slow for ab...
1	2	adjudicated	poor	Recommended by Don Katz. Avail for free in Dec...
2	3	adjudicated	good	A fun, fast paced science fiction thriller. I ...
3	4	adjudicated	poor	Recommended reading to understand what is goin...
4	5	adjudicated	average	I really enjoyed this book, and there is a lot...

```
# Read the adjudicated.txt file into a DataFrame without a header
df = pd.read_csv('adjudicated.txt', sep='\t', header=None)

# Shuffle the DataFrame
df_shuffled = df.sample(frac=1, random_state=42)  # Random_state for reproducibility

# Split the shuffled DataFrame into train, dev, and test sets
train = df_shuffled[:300]
dev = df_shuffled[300:400]
test = df_shuffled[400:]

# Write each split to a separate text file
train.to_csv('train.txt', sep='\t', header=None, index=False)
dev.to_csv('dev.txt', sep='\t', header=None, index=False)
test.to_csv('test.txt', sep='\t', header=None, index=False)
```

```
df_labels_count = df_shuffled[2].value_counts().sort_index()
print("df labels count:")
print(df_labels_count)
```

```
df labels count:
2
average      208
excellent     52
good         182
poor          58
Name: count, dtype: int64
```

```
train_labels_count = train[2].value_counts().sort_index()
dev_labels_count = dev[2].value_counts().sort_index()
test_labels_count = test[2].value_counts().sort_index()

print("Train labels count:")
print(train_labels_count)

print("\nDev labels count:")
print(dev_labels_count)

print("\nTest labels count:")
print(test_labels_count)
```

```
Train labels count:
2
average      119
excellent     28
good         120
poor          33
Name: count, dtype: int64

Dev labels count:
2
average      46
excellent    11
good         32
poor         11
Name: count, dtype: int64

Test labels count:
2
average      43
excellent    13
good         30
poor         14
Name: count, dtype: int64
```