DonorsChoose

DonorsChoose.org receives hundreds of thousands of project proposals each year for classroom projects in need of funding. Right now, a large number of volunteers is needed to manually screen each submission before it's approved to be posted on the DonorsChoose.org website.

Next year, DonorsChoose.org expects to receive close to 500,000 project proposals. As a result, there are three main problems they need to solve:

- How to scale current manual processes and resources to screen 500,000 projects so that they can be posted as quickly and as efficiently as possible
- How to increase the consistency of project vetting across different volunteers to improve the experience for teachers
- How to focus volunteer time on the applications that need the most assistance

The goal of the competition is to predict whether or not a DonorsChoose.org project proposal submitted by a teacher will be approved, using the text of project descriptions as well as additional metadata about the project, teacher, and school. DonorsChoose.org can then use this information to identify projects most likely to need further review before approval.

About the DonorsChoose Data Set

The `train.csv` data set provided by DonorsChoose contains the following features:

Feature	Descri
`project_id`	A unique identifier for the proposed project. **Example:** `p03`
`project_title`	Title of the project. **Exam** • `Art Will Make You Ha` • `First Grade`
`project_grade_category`	Grade level of students for which the project is targeted. One of the foll enumerated va • `Grades Pr` • `Grades` • `Grades` • `Grades`
`project_subject_categories`	One or more (comma-separated) subject categories for the project fro following enumerated list of va • `Applied Lear` • `Care & Hu` • `Health & Sp` • `History & Ci` • `Literacy & Lang` • `Math & Sci` • `Music & The` • `Special N` • `Wa` **Exam** • `Music & The` • `Literacy & Language, Math & Sci`
`school_state`	State where school is located ([Two-letter U.S. postal (https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Postal_co](https://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations#Postal_co) **Example**
`project_subject_subcategories`	One or more (comma-separated) subject subcategories for the pr **Exam** • `Lite` • `Literature & Writing, Social Scie`
`project_resource_summary`	An explanation of the resources needed for the project. **Exan** • `My students need hands on literacy materials to mar sensory ne`
`project_essay_1`	First application e
`project_essay_2`	Second application e
`project_essay_3`	Third application e
`project_essay_4`	Fourth application e

Feature	Descri
project_submitted_datetime	Datetime when project application was submitted. **Example:** 2016-0 12:43:56
teacher_id	A unique identifier for the teacher of the proposed project. **Exa** bdf8baa8fedef6bfeec7ae4ff1c1
teacher_prefix	Teacher's title. One of the following enumerated va • • • • • • Teac
teacher_number_of_previously_posted_projects	Number of project applications previously submitted by the same te **Exampl**

* See the section **Notes on the Essay Data** for more details about these features.

Additionally, the `resources.csv` data set provides more data about the resources required for each project. Each line in this file represents a resource required by a project:

Feature	Description
id	A `project_id` value from the `train.csv` file. **Example:** `p036502`
description	Desciption of the resource. **Example:** `Tenor Saxophone Reeds, Box of 25`
quantity	Quantity of the resource required. **Example:** `3`
price	Price of the resource required. **Example:** `9.95`

Note: Many projects require multiple resources. The `id` value corresponds to a `project_id` in train.csv, so you use it as a key to retrieve all resources needed for a project:

The data set contains the following label (the value you will attempt to predict):

Label	Description
`project_is_approved`	A binary flag indicating whether DonorsChoose approved the project. A value of `0` indicates the project was not approved, and a value of `1` indicates the project was approved.

Notes on the Essay Data

Prior to May 17, 2016, the prompts for the essays were as follows:

- __project_essay_1:__ "Introduce us to your classroom"
- __project_essay_2:__ "Tell us more about your students"
- __project_essay_3:__ "Describe how your students will use the materials you're requesting"
- __project_essay_3:__ "Close by sharing why your project will make a difference"

Starting on May 17, 2016, the number of essays was reduced from 4 to 2, and the prompts for the first 2 essays were changed to the following:

- __project_essay_1:__ "Describe your students: What makes your students special? Specific details about their background, your neighborhood, and your school are all helpful."
- __project_essay_2:__ "About your project: How will these materials make a difference in your students' learning and improve their school lives?"

For all projects with project_submitted_datetime of 2016-05-17 and later, the values of project_essay_3 and project_essay_4 will be NaN.

In [1]:
```
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")

import sqlite3
import pandas as pd
import numpy as np
import nltk
import string
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import confusion_matrix
from sklearn import metrics
from sklearn.metrics import roc_curve, auc
from nltk.stem.porter import PorterStemmer

import re
# Tutorial about Python regular expressions: https://pymotw.com/2/re/
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

from gensim.models import Word2Vec
from gensim.models import KeyedVectors
import pickle

from tqdm import tqdm
import os

from chart_studio import plotly
import plotly.offline as offline
import plotly.graph_objs as go
offline.init_notebook_mode()
from collections import Counter
```

1.1 Reading Data

In [2]:
```
project_data = pd.read_csv('train_data.csv', nrows = 80000) # Taking 80000 dat
apoints
resource_data = pd.read_csv('resources.csv')
```

In [3]:
```
print("Number of data points in train data", project_data.shape)
print('-'*50)
print("The attributes of data :", project_data.columns.values)
```

```
Number of data points in train data (80000, 17)
--------------------------------------------------
The attributes of data : ['Unnamed: 0' 'id' 'teacher_id' 'teacher_prefix' 'sc
hool_state'
 'project_submitted_datetime' 'project_grade_category'
 'project_subject_categories' 'project_subject_subcategories'
 'project_title' 'project_essay_1' 'project_essay_2' 'project_essay_3'
 'project_essay_4' 'project_resource_summary'
 'teacher_number_of_previously_posted_projects' 'project_is_approved']
```

In [4]:
```
# how to replace elements in list python: https://stackoverflow.com/a/2582163/
4084039
cols = ['Date' if x=='project_submitted_datetime' else x for x in list(project
_data.columns)]


#sort dataframe based on time pandas python: https://stackoverflow.com/a/49702
492/4084039
project_data['Date'] = pd.to_datetime(project_data['project_submitted_datetim
e'])
project_data.drop('project_submitted_datetime', axis=1, inplace=True)
project_data.sort_values(by=['Date'], inplace=True)


# how to reorder columns pandas python: https://stackoverflow.com/a/13148611/4
084039
project_data = project_data[cols]


project_data.head(2)
```

Out[4]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	I
55660	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA	2 0 00:2
76127	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms.	UT	2 0 00:3

In [5]:
```
print("Number of data points in train data", resource_data.shape)
print(resource_data.columns.values)
resource_data.head(2)
```

```
Number of data points in train data (1541272, 4)
['id' 'description' 'quantity' 'price']
```

Out[5]:

	id	description	quantity	price
0	p233245	LC652 - Lakeshore Double-Space Mobile Drying Rack	1	149.00
1	p069063	Bouncy Bands for Desks (Blue support pipes)	3	14.95

1.2 preprocessing of `project_subject_categories`

In [6]:
```
catogories = list(project_data['project_subject_categories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-in-python
cat_list = []
for i in catogories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp+=j.strip()+" " #" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_') # we are replacing the & value into
    cat_list.append(temp.strip())

project_data['clean_categories'] = cat_list
project_data.drop(['project_subject_categories'], axis=1, inplace=True)

from collections import Counter
my_counter = Counter()
for word in project_data['clean_categories'].values:
    my_counter.update(word.split())

cat_dict = dict(my_counter)
sorted_cat_dict = dict(sorted(cat_dict.items(), key=lambda kv: kv[1]))
```

1.3 preprocessing of `project_subject_subcategories`

In [7]:

```
sub_catogories = list(project_data['project_subject_subcategories'].values)
# remove special characters from list of strings python: https://stackoverflo
w.com/a/47301924/4084039

# https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
# https://stackoverflow.com/questions/23669024/how-to-strip-a-specific-word-fr
om-a-string
# https://stackoverflow.com/questions/8270092/remove-all-whitespace-in-a-strin
g-in-python

sub_cat_list = []
for i in sub_catogories:
    temp = ""
    # consider we have text like this "Math & Science, Warmth, Care & Hunger"
    for j in i.split(','): # it will split it in three parts ["Math & Scienc
e", "Warmth", "Care & Hunger"]
        if 'The' in j.split(): # this will split each of the catogory based on
space "Math & Science"=> "Math","&", "Science"
            j=j.replace('The','') # if we have the words "The" we are going to
replace it with ''(i.e removing 'The')
        j = j.replace(' ','') # we are placeing all the ' '(space) with ''(emp
ty) ex:"Math & Science"=>"Math&Science"
        temp +=j.strip()+" "#" abc ".strip() will return "abc", remove the tra
iling spaces
        temp = temp.replace('&','_')
    sub_cat_list.append(temp.strip())

project_data['clean_subcategories'] = sub_cat_list
project_data.drop(['project_subject_subcategories'], axis=1, inplace=True)

# count of all the words in corpus python: https://stackoverflow.com/a/2289859
5/4084039
my_counter = Counter()
for word in project_data['clean_subcategories'].values:
    my_counter.update(word.split())

sub_cat_dict = dict(my_counter)
sorted_sub_cat_dict = dict(sorted(sub_cat_dict.items(), key=lambda kv: kv[1]))
```

1.3 Text preprocessing

In [8]:

```
# merge two column text dataframe:
project_data["essay"] = project_data["project_essay_1"].map(str) +\
                        project_data["project_essay_2"].map(str) + \
                        project_data["project_essay_3"].map(str) + \
                        project_data["project_essay_4"].map(str)
```

In [9]:
```
project_data.head(2)
```

Out[9]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	
55660	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs.	CA	2 0 00:2
76127	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms.	UT	2 0 00:3

1.4.2.3 Using Pretrained Models: TFIDF weighted W2V

In [10]:
```
# printing some random reviews
print(project_data['essay'].values[0])
print("="*50)
print(project_data['essay'].values[150])
print("="*50)
print(project_data['essay'].values[1000])
print("="*50)
print(project_data['essay'].values[20000])
print("="*50)
print(project_data['essay'].values[79999])
print("="*50)
```

```
I have been fortunate enough to use the Fairy Tale STEM kits in my classroom
as well as the STEM journals, which my students really enjoyed.  I would love
to implement more of the Lakeshore STEM kits in my classroom for the next sch
ool year as they provide excellent and engaging STEM lessons.My students come
from a variety of backgrounds, including language and socioeconomic status.
Many of them don't have a lot of experience in science and engineering and th
ese kits give me the materials to provide these exciting opportunities for my
students.Each month I try to do several science or STEM/STEAM projects.  I wo
uld use the kits and robot to help guide my science instruction in engaging a
nd meaningful ways.  I can adapt the kits to my current language arts pacing
guide where we already teach some of the material in the kits like tall tales
(Paul Bunyan) or Johnny Appleseed.  The following units will be taught in the
next school year where I will implement these kits: magnets, motion, sink vs.
float, robots.  I often get to these units and don't know If I am teaching th
e right way or using the right materials.    The kits will give me additional
ideas, strategies, and lessons to prepare my students in science.It is challe
nging to develop high quality science activities.  These kits give me the mat
erials I need to provide my students with science activities that will go alo
ng with the curriculum in my classroom.  Although I have some things (like ma
gnets) in my classroom, I don't know how to use them effectively.  The kits w
ill provide me with the right amount of materials and show me how to use them
in an appropriate way.
==================================================
\"Why can't I play the drums, Mr. Reyes?\"  I hear this one often.  I want my
students to experience \"percussion\" as opposed to \"the drums.\"  We need s
ome percussion instruments in order to help make this dream come to fruition.
We have a diverse population with almost entirely “minority students” in atte
ndance, including many English Learners (several who barely speak any Englis
h). We have great staff members that love the kids and support them in all as
pects of their education. Students’ safety is the primary interest at our sch
ool.  We hope to provide a well-rounded music education where band is “cool”
and kids look forward to coming to class. We are finding ourselves in need of
instruments to help keep the momentum of our program going.My students will b
e able to experience more advanced pieces of music with this instrument.  I a
m asking for a high quality instrument so that it may be used and taken care
of over time.  Eventually, I’d love to start a percussion ensemble.  This inv
estment will greatly help the percussion section at my school!My students com
e from very troubled homes and they deal with stress that people shouldn’t ha
ve to experience. They know adversity more than the average person and have t
o over come it in whatever ways they can manage. Music can be an amazing outl
et for these kids. Your generous support could help change a life for a young
troubled youth. Thank you for your help!
==================================================
I have over 80 students on my ESL roster that need my services during the sch
ool year.They come in at varying levels of English acquisition, so differenti
ation is very important in my classroom.Technology is only becoming more impo
rtant, so giving my students this chance to be hands-on is wonderful.My schoo
l has over 740 students; about 80% are of a minority group and roughly 60% of
them are English Language Learners. There are more than 7 languages represent
ed, some mainstream like Spanish or Vietnamese, and some indigenous like Mam.
The neighborhood where my school is located is very low income, with some fam
ilies being in poverty or homeless.With all of these things working against t
hem, you will not meet a group of students who are more dedicated to learnin
g.They are excited to try new things.My school works very hard to give the st
udents as much access to print (good literature) and hands-on activities as w
e can.If my project is funded, my students will be able to use the iPad Minis
during class time in several different ways. They will use them to look up wo
```

```
rds they don't understand, or to find synonyms and antonyms to help broaden t
heir vocabularies. There are so many apps available that help English Languag
e Learners practice and apply what they are learning in their classrooms. My
students will be able to use these apps as hands-on practice when our lessons
are complete, or as a reward for a job well done. As the majority of our stud
ents are low income, being able to work with technology in school can help th
em to become more future-ready.This project will make a difference in my clas
sroom because it gives my students another avenue for learning. It gets techn
ology into their hands that they don't have access to normally, making them m
ore future-ready. It will give them a hands-on learning tool to make the mate
rial more meaningful to them. It will give them a sense of ownership over the
material and be an encouragement for learning a new language.
==================================================
In my classroom, I have the opportunity to make a difference in the lives of
children who face challenges associated with limited English language profici
ency and socioeconomic status. Most of the students at my school walk into ou
r classrooms after only being in the country for a few days-Imagine walking i
nto a room and seeing a book for the first time.. or a pencil. A smiling teac
her is talking to you in a language you don't understand-Many of our other st
udents are homeless or being raised in a single parent household and receive
a free lunch based on their socioeconomic status. These are the students I te
ach every day.\r\nDespite these challenges, I believe that every single stude
nt can learn, grow and accomplish amazing things. I can not control what happ
ens outside of the classroom, however, I can certainly control what happens i
nside the classroom. I will do everything in my power to give my students the
best education and most positive and nurturing learning environment. I am hop
eful to inspire our earliest learners to continue on a path to academic excel
lence.My job is to not only teach my students how to read, but to also get th
em excited about it! A major part in motivating the students is allowing them
to choose where they sit in the classroom as they read-no one wants to sit at
their desk every day! The problem with this, however, is our current classroo
m library has two pillows and one bean bag. This makes the option of getting
comfy while reading around the room very limited. \r\nDonations for this proj
ect will allow more of my students the option to be comfortable while they re
ad. My students will be able to sit on the soft pillows not only in the libra
ry, but also around the room. My students will also enjoy the soft plush anim
als to snuggle as they read. With your help, I can transform our reading time
into something every student will enjoy!nannan
==================================================
My first graders are eager to learn about the world around them.  They come t
o school each day full of enthusiasm and genuinely love learning. \r\n\r\nOur
diverse class includes students from a variety of cultural and economic backg
rounds.  Many come from homes where parents can't afford or simply don't know
the importance of books, so it is important to me to provide an environment t
hat is rich in literature so that students learn to love reading.  I want my
students to be lifelong learners, and reading is the best way!I have used the
se magazines in the past, and kids absolutely LOVE them!! The topics are of h
igh interest for children and always correspond to real world issues that are
important for kids to learn. The subscription also includes online resources
such as videos, printable worksheets, and skill-based games. \r\n\r\nThese ma
terials will expose students to rigorous and interesting nonfiction text that
will spark their curiosity about the world around them.  The topics allow me
to teach the nonfiction text standards using interesting materials.  They alw
ays lead to engaging discussions and inspire students to find additional info
rmation about the various topics.nannan
==================================================
```

In [11]:
```
# https://stackoverflow.com/a/47091490/4084039
import re

def decontracted(phrase):
    # specific
    phrase = re.sub(r"won't", "will not", phrase)
    phrase = re.sub(r"can\'t", "can not", phrase)

    # general
    phrase = re.sub(r"n\'t", " not", phrase)
    phrase = re.sub(r"\'re", " are", phrase)
    phrase = re.sub(r"\'s", " is", phrase)
    phrase = re.sub(r"\'d", " would", phrase)
    phrase = re.sub(r"\'ll", " will", phrase)
    phrase = re.sub(r"\'t", " not", phrase)
    phrase = re.sub(r"\'ve", " have", phrase)
    phrase = re.sub(r"\'m", " am", phrase)
    return phrase
```

In [12]:
```
sent = decontracted(project_data['essay'].values[20000])
print(sent)
print("="*50)
```

```
In my classroom, I have the opportunity to make a difference in the lives of
children who face challenges associated with limited English language profici
ency and socioeconomic status. Most of the students at my school walk into ou
r classrooms after only being in the country for a few days-Imagine walking i
nto a room and seeing a book for the first time.. or a pencil. A smiling teac
her is talking to you in a language you do not understand-Many of our other s
tudents are homeless or being raised in a single parent household and receive
a free lunch based on their socioeconomic status. These are the students I te
ach every day.\r\nDespite these challenges, I believe that every single stude
nt can learn, grow and accomplish amazing things. I can not control what happ
ens outside of the classroom, however, I can certainly control what happens i
nside the classroom. I will do everything in my power to give my students the
best education and most positive and nurturing learning environment. I am hop
eful to inspire our earliest learners to continue on a path to academic excel
lence.My job is to not only teach my students how to read, but to also get th
em excited about it! A major part in motivating the students is allowing them
to choose where they sit in the classroom as they read-no one wants to sit at
their desk every day! The problem with this, however, is our current classroo
m library has two pillows and one bean bag. This makes the option of getting
comfy while reading around the room very limited. \r\nDonations for this proj
ect will allow more of my students the option to be comfortable while they re
ad. My students will be able to sit on the soft pillows not only in the libra
ry, but also around the room. My students will also enjoy the soft plush anim
als to snuggle as they read. With your help, I can transform our reading time
into something every student will enjoy!nannan
==================================================
```

In [13]:
```
# \r \n \t remove from string python: http://texthandler.com/info/remove-line-breaks-python/
sent = sent.replace('\\r', ' ')
sent = sent.replace('\\"', ' ')
sent = sent.replace('\\n', ' ')
print(sent)
```

```
In my classroom, I have the opportunity to make a difference in the lives of
children who face challenges associated with limited English language profici
ency and socioeconomic status. Most of the students at my school walk into ou
r classrooms after only being in the country for a few days-Imagine walking i
nto a room and seeing a book for the first time.. or a pencil. A smiling teac
her is talking to you in a language you do not understand-Many of our other s
tudents are homeless or being raised in a single parent household and receive
a free lunch based on their socioeconomic status. These are the students I te
ach every day.  Despite these challenges, I believe that every single student
can learn, grow and accomplish amazing things. I can not control what happens
outside of the classroom, however, I can certainly control what happens insid
e the classroom. I will do everything in my power to give my students the bes
t education and most positive and nurturing learning environment. I am hopefu
l to inspire our earliest learners to continue on a path to academic excellen
ce.My job is to not only teach my students how to read, but to also get them
excited about it! A major part in motivating the students is allowing them to
choose where they sit in the classroom as they read-no one wants to sit at th
eir desk every day! The problem with this, however, is our current classroom
library has two pillows and one bean bag. This makes the option of getting co
mfy while reading around the room very limited.   Donations for this project
will allow more of my students the option to be comfortable while they read.
My students will be able to sit on the soft pillows not only in the library,
but also around the room. My students will also enjoy the soft plush animals
to snuggle as they read. With your help, I can transform our reading time int
o something every student will enjoy!nannan
```

In [14]:
```
#remove spacial character: https://stackoverflow.com/a/5843547/4084039
sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
print(sent)
```

```
In my classroom I have the opportunity to make a difference in the lives of c
hildren who face challenges associated with limited English language proficie
ncy and socioeconomic status Most of the students at my school walk into our 
classrooms after only being in the country for a few days Imagine walking int
o a room and seeing a book for the first time or a pencil A smiling teacher i
s talking to you in a language you do not understand Many of our other studen
ts are homeless or being raised in a single parent household and receive a fr
ee lunch based on their socioeconomic status These are the students I teach e
very day Despite these challenges I believe that every single student can lea
rn grow and accomplish amazing things I can not control what happens outside 
of the classroom however I can certainly control what happens inside the clas
sroom I will do everything in my power to give my students the best education
and most positive and nurturing learning environment I am hopeful to inspire 
our earliest learners to continue on a path to academic excellence My job is 
to not only teach my students how to read but to also get them excited about 
it A major part in motivating the students is allowing them to choose where t
hey sit in the classroom as they read no one wants to sit at their desk every
day The problem with this however is our current classroom library has two pi
llows and one bean bag This makes the option of getting comfy while reading a
round the room very limited Donations for this project will allow more of my 
students the option to be comfortable while they read My students will be abl
e to sit on the soft pillows not only in the library but also around the room
My students will also enjoy the soft plush animals to snuggle as they read Wi
th your help I can transform our reading time into something every student wi
ll enjoy nannan
```

In [15]:

```
# https://gist.github.com/sebleier/554280
# we are removing the words from the stop words list: 'no', 'nor', 'not'
stopwords= ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you'
, "you're", "you've",\
            "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he'
, 'him', 'his', 'himself', \
            'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'it
self', 'they', 'them', 'their',\
            'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 't
hat', "that'll", 'these', 'those', \
            'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have',
'has', 'had', 'having', 'do', 'does', \
            'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'becau
se', 'as', 'until', 'while', 'of', \
            'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into',
'through', 'during', 'before', 'after',\
            'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on',
'off', 'over', 'under', 'again', 'further',\
            'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'a
ll', 'any', 'both', 'each', 'few', 'more',\
            'most', 'other', 'some', 'such', 'only', 'own', 'same', 'so', 'tha
n', 'too', 'very', \
            's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "shoul
d've", 'now', 'd', 'll', 'm', 'o', 're', \
            've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn',
"didn't", 'doesn', "doesn't", 'hadn',\
            "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'm
a', 'mightn', "mightn't", 'mustn',\
            "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shoul
dn't", 'wasn', "wasn't", 'weren', "weren't", \
            'won', "won't", 'wouldn', "wouldn't"]
```

In [16]:

```
# Combining all the above stundents
from tqdm import tqdm
preprocessed_essays = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['essay'].values):
    sent = decontracted(sentance)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    # https://gist.github.com/sebleier/554280
    sent = ' '.join(e for e in sent.split() if e.lower() not in stopwords)
    preprocessed_essays.append(sent.lower().strip())
```

```
100%|██████████████████████████████████████████
█| 80000/80000 [00:38<00:00, 2087.22it/s]
```

In [17]:
```
# after preprocesing
preprocessed_essays[20000]
```

Out[17]:
```
'classroom opportunity make difference lives children face challenges associa
ted limited english language proficiency socioeconomic status students school
walk classrooms country days imagine walking room seeing book first time penc
il smiling teacher talking language not understand many students homeless rai
sed single parent household receive free lunch based socioeconomic status stu
dents teach every day despite challenges believe every single student learn g
row accomplish amazing things not control happens outside classroom however c
ertainly control happens inside classroom everything power give students best
education positive nurturing learning environment hopeful inspire earliest le
arners continue path academic excellence job not teach students read also get
excited major part motivating students allowing choose sit classroom read no
one wants sit desk every day problem however current classroom library two pi
llows one bean bag makes option getting comfy reading around room limited don
ations project allow students option comfortable read students able sit soft
pillows not library also around room students also enjoy soft plush animals s
nuggle read help transform reading time something every student enjoy nannan'
```

1.4 Preprocessing of `project_title`

In [18]:
```
# similarly you can preprocess the titles also
def preprocess_text_func(text_data):
    sent = decontracted(text_data)
    sent = sent.replace('\\r', ' ')
    sent = sent.replace('\\"', ' ')
    sent = sent.replace('\\n', ' ')
    sent = re.sub('[^A-Za-z0-9]+', ' ', sent)
    sent = ' '.join(e for e in sent.split() if e not in stopwords)
    return sent.lower()
```

In [19]:
```
preprocessed_titles = []
# tqdm is for printing the status bar
for sentance in tqdm(project_data['project_title'].values):
    preprocessed_titles.append(preprocess_text_func(sentance))
```

```
100%|██████████████████████████████████████████████████
█| 80000/80000 [00:01<00:00, 48061.09it/s]
```

In [20]:
```
print(preprocessed_titles[5:12])
```

```
['breakout box ignite engagement', '21st century learning multimedia', 'ipad
learners', 'a flexible classroom flexible minds', 'make powerful movies', 'ro
bots taking 2nd grade', 'time kids to learn about science']
```

In [21]:
```
print(project_data["project_title"].values[5:12])
```

```
['Breakout Box to Ignite Engagement!'
 '21st Century Learning with Multimedia' 'iPad for Learners'
 'A flexible classroom for flexible minds!' 'Make Powerful Movies!!'
 'Robots are Taking over 2nd Grade'
 'Time for Kids....To Learn About Science and more!']
```

1.5 Preparing data for models

In [22]:
```
project_data.columns
```

Out[22]:
```
Index(['Unnamed: 0', 'id', 'teacher_id', 'teacher_prefix', 'school_state',
       'Date', 'project_grade_category', 'project_title', 'project_essay_1',
       'project_essay_2', 'project_essay_3', 'project_essay_4',
       'project_resource_summary',
       'teacher_number_of_previously_posted_projects', 'project_is_approved',
       'clean_categories', 'clean_subcategories', 'essay'],
      dtype='object')
```

we are going to consider

```
- school_state : categorical data
- clean_categories : categorical data
- clean_subcategories : categorical data
- project_grade_category : categorical data
- teacher_prefix : categorical data

- project_title : text data
- text : text data
- project_resource_summary: text data (optinal)

- quantity : numerical (optinal)
- teacher_number_of_previously_posted_projects : numerical
- price : numerical
```

1.5.1 Vectorizing Categorical data

- https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/ (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/handling-categorical-and-numerical-features/)

In [23]:
```
# we use count vectorizer to convert the values into one
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer(vocabulary=list(sorted_cat_dict.keys()), lowercas
e=False, binary=True)
categories_one_hot = vectorizer.fit_transform(project_data['clean_categories']
.values)
print(vectorizer.get_feature_names())
print("Shape of matrix after one hot encodig ",categories_one_hot.shape)
```

```
['Warmth', 'Care_Hunger', 'History_Civics', 'Music_Arts', 'AppliedLearning',
'SpecialNeeds', 'Health_Sports', 'Math_Science', 'Literacy_Language']
Shape of matrix after one hot encodig  (80000, 9)
```

In [24]:
```
# we use count vectorizer to convert the values into one
vectorizer = CountVectorizer(vocabulary=list(sorted_sub_cat_dict.keys()), lowe
rcase=False, binary=True)
sub_categories_one_hot = vectorizer.fit_transform(project_data['clean_subcateg
ories'].values)
print(vectorizer.get_feature_names())
print("Shape of matrix after one hot encodig ",sub_categories_one_hot.shape)
```

```
['Economics', 'CommunityService', 'FinancialLiteracy', 'ParentInvolvement',
'Civics_Government', 'Extracurricular', 'ForeignLanguages', 'NutritionEducati
on', 'Warmth', 'Care_Hunger', 'SocialSciences', 'PerformingArts', 'CharacterE
ducation', 'TeamSports', 'Other', 'College_CareerPrep', 'History_Geography',
'Music', 'Health_LifeScience', 'EarlyDevelopment', 'ESL', 'Gym_Fitness', 'Env
ironmentalScience', 'VisualArts', 'Health_Wellness', 'AppliedSciences', 'Spec
ialNeeds', 'Literature_Writing', 'Mathematics', 'Literacy']
Shape of matrix after one hot encodig  (80000, 30)
```

In [25]:
```
# you can do the similar thing with state, teacher_prefix and project_grade_ca
tegory also
# function to perform one hot encoding
def perform_one_hot_encoding(listdata, category,fillnan_value=""):
    vectorizer =  CountVectorizer(vocabulary=listdata, lowercase=False, binary
=True)
    vectorizer.fit(project_data[category].fillna(fillnan_value).values)
    print(vectorizer.get_feature_names())
    print("="*50)
    return vectorizer.transform(project_data[category].fillna(fillnan_value).v
alues)
```

In [26]:
```
# One hot encoding for school state
countries_list = sorted(project_data["school_state"].value_counts().keys())
school_state_one_hot = perform_one_hot_encoding(countries_list, "school_state"
)
print("Shape of matrix after one hot encodig ",school_state_one_hot.shape)
```

```
['AK', 'AL', 'AR', 'AZ', 'CA', 'CO', 'CT', 'DC', 'DE', 'FL', 'GA', 'HI', 'I
A', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA', 'MD', 'ME', 'MI', 'MN', 'MO',
'MS', 'MT', 'NC', 'ND', 'NE', 'NH', 'NJ', 'NM', 'NV', 'NY', 'OH', 'OK', 'OR',
'PA', 'RI', 'SC', 'SD', 'TN', 'TX', 'UT', 'VA', 'VT', 'WA', 'WI', 'WV', 'WY']
==================================================
Shape of matrix after one hot encodig  (80000, 51)
```

In [27]:
```
# Project_Grade_Category - replacing hyphens, spaces with Underscores
project_data['project_grade_category'] = project_data['project_grade_category'
].map({'Grades PreK-2': 'Grades_PreK_2',

'Grades 6-8' : 'Grades_6_8',

'Grades 3-5' : 'Grades_3_5',

'Grades 9-12' : 'Grades_9_12'})
project_data['teacher_prefix'] = project_data['teacher_prefix'].map({'Mrs.':
'Mrs', 'Ms.': 'Ms', 'Mr.' : 'Mr',
                                                                       'Teacher'
: 'Teacher', 'Dr.' : 'Dr'})
```

In [28]:
```
project_data['project_grade_category'].head(5)
```

Out[28]:
```
55660    Grades_PreK_2
76127       Grades_3_5
51140    Grades_PreK_2
473      Grades_PreK_2
41558       Grades_3_5
Name: project_grade_category, dtype: object
```

In [29]:
```
project_data['teacher_prefix'].head(5)
```

Out[29]:
```
55660    Mrs
76127     Ms
51140    Mrs
473      Mrs
41558    Mrs
Name: teacher_prefix, dtype: object
```

In [30]:
```
project_data['teacher_prefix'].isnull().values.any() # since there are null va
lues replacing them with most common value Mrs
```

Out[30]:
```
True
```

In [31]:
```
project_data["teacher_prefix"].fillna("Mrs", inplace=True)
```

In [32]:
```
project_data['teacher_prefix'].isnull().values.any() # No more Null Values
```

Out[32]:
```
False
```

In [33]:
```
# One hot encoding for teacher_prefix
teacher_prefix_list = sorted(project_data["teacher_prefix"].value_counts().key
s())
print (teacher_prefix_list)
teacher_prefix_one_hot = perform_one_hot_encoding(teacher_prefix_list, "teache
r_prefix", "Mrs.")
print("Shape of matrix after one hot encodig ",teacher_prefix_one_hot.shape)
```

```
['Dr', 'Mr', 'Mrs', 'Ms', 'Teacher']
['Dr', 'Mr', 'Mrs', 'Ms', 'Teacher']
==================================================
Shape of matrix after one hot encodig  (80000, 5)
```

In [34]:
```
# One hot encoding for project_grade_category
grade_list = sorted(project_data["project_grade_category"].value_counts().keys
())
grade_one_hot = perform_one_hot_encoding(grade_list, "project_grade_category")
print("Shape of matrix after one hot encodig ",grade_one_hot.shape)
```

```
['Grades_3_5', 'Grades_6_8', 'Grades_9_12', 'Grades_PreK_2']
==================================================
Shape of matrix after one hot encodig  (80000, 4)
```

1.5.2 Vectorizing Text data

1.5.2.1 Bag of words

In [35]:
```
# We are considering only the words which appeared in at least 10 documents(ro
ws or projects).
vectorizer = CountVectorizer(min_df=10)
text_bow = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_bow.shape)
```

```
Shape of matrix after one hot encodig  (80000, 14627)
```

In [36]:
```
# you can vectorize the title also
# before you vectorize the title make sure you preprocess it
vectorizer_titles = CountVectorizer(min_df=10)
text_bow_titles = vectorizer_titles.fit_transform(preprocessed_titles)
print("Shape of matrix after one hot encodig ",text_bow_titles.shape)
```

```
Shape of matrix after one hot encodig  (80000, 2740)
```

1.5.2.2 TFIDF vectorizer

In [37]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10)
text_tfidf = vectorizer.fit_transform(preprocessed_essays)
print("Shape of matrix after one hot encodig ",text_tfidf.shape)
```

```
Shape of matrix after one hot encodig  (80000, 14627)
```

In [38]:
```
# TFIDF  Vectorizer for Preprocessed Title
vectorizer_titles = TfidfVectorizer(min_df=10)
text_tfidf_titles = vectorizer_titles.fit_transform(preprocessed_titles)
print("Shape of matrix after one hot encodig ",text_tfidf_titles.shape)
```

```
Shape of matrix after one hot encodig  (80000, 2740)
```

1.5.2.3 Using Pretrained Models: Avg W2V

In [ ]:
```
'''
# Reading glove vectors in python: https://stackoverflow.com/a/38230349/408403
9
def loadGloveModel(gloveFile):
    print ("Loading Glove Model")
    f = open(gloveFile,'r', encoding="utf8")
    model = {}
    for line in tqdm(f):
        splitLine = line.split()
        word = splitLine[0]
        embedding = np.array([float(val) for val in splitLine[1:]])
        model[word] = embedding
    print ("Done.",len(model)," words loaded!")
    return model
model = loadGloveModel('glove.42B.300d.txt')

# ============================
Output:

Loading Glove Model
1917495it [06:32, 4879.69it/s]
Done. 1917495  words loaded!

# ============================

words = []
for i in preproced_texts:
    words.extend(i.split(' '))

for i in preproced_titles:
    words.extend(i.split(' '))
print("all the words in the coupus", len(words))
words = set(words)
print("the unique words in the coupus", len(words))

inter_words = set(model.keys()).intersection(words)
print("The number of words that are present in both glove vectors and our coup
us", \
      len(inter_words),"(",np.round(len(inter_words)/len(words)*100,3),"%)")

words_courpus = {}
words_glove = set(model.keys())
for i in words:
    if i in words_glove:
        words_courpus[i] = model[i]
print("word 2 vec length", len(words_courpus))


# stronging variables into pickle files python: http://www.jessicayung.com/how
-to-use-pickle-to-save-and-load-variables-in-python/

import pickle
with open('glove_vectors', 'wb') as f:
    pickle.dump(words_courpus, f)
```

```
'''
```

In [39]:
```
# stronging variables into pickle files python: http://www.jessicayung.com/how
-to-use-pickle-to-save-and-load-variables-in-python/
# make sure you have the glove_vectors file
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

In [40]:
```
# average Word2Vec
# compute average word2vec for each review.
avg_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in this
list
for sentence in tqdm(preprocessed_essays): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors.append(vector)

print(len(avg_w2v_vectors))
print(len(avg_w2v_vectors[0]))
```

```
100%|████████████████████████████████████████████████████████████████████████
██| 80000/80000 [00:16<00:00, 4719.02it/s]

80000
300
```

In [41]:
```
# average Word2Vec for Preprocessed Titles
avg_w2v_vectors_titles = []; # the avg-w2v for each project_title is stored in
this list
for sentence in tqdm(preprocessed_titles): # for each project_title
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_titles.append(vector)

print(len(avg_w2v_vectors_titles))
print(len(avg_w2v_vectors_titles[0]))
```

```
100%|████████████████████████████████████████████████████████████████████████
█| 80000/80000 [00:00<00:00, 81851.02it/s]

80000
300
```

1.5.2.3 Using Pretrained Models: TFIDF weighted W2V

In [42]:
```
# S = ["abc def pqr", "def def def abc", "pqr pqr def"]
tfidf_model = TfidfVectorizer()
tfidf_model.fit(preprocessed_essays)
# we are converting a dictionary with word as a key, and the idf as a value
dictionary = dict(zip(tfidf_model.get_feature_names(), list(tfidf_model.idf_
)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [43]:
```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors = []; # the avg-w2v for each sentence/review is stored in th
is list
for sentence in tqdm(preprocessed_essays): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/revie
w
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf v
alue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors.append(vector)

print(len(tfidf_w2v_vectors))
print(len(tfidf_w2v_vectors[0]))
```

```
100%|██████████████████████████████████████████████████████████████████
███| 80000/80000 [01:58<00:00, 672.90it/s]

80000
300
```

In [44]:
```
# Similarly you can vectorize for title also
tfidf_w2v_vectors_titles = []; # the avg-w2v for each project_title is stored in this list
for sentence in tqdm(preprocessed_titles): # for each project_title
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_titles.append(vector)

print(len(tfidf_w2v_vectors_titles))
print(len(tfidf_w2v_vectors_titles[0]))
```

```
100%|██████████████████████████████████████████████████████████████████████
█| 80000/80000 [00:01<00:00, 40167.22it/s]

80000
300
```

1.5.3 Vectorizing Numerical features

In [45]:
```
price_data = resource_data.groupby('id').agg({'price':'sum', 'quantity':'sum'}).reset_index()
project_data = pd.merge(project_data, price_data, on='id', how='left')
```

In [46]:
```
# check this one: https://www.youtube.com/watch?v=0HOqOcln3Z4&t=530s
# standardization sklearn: https://scikit-learn.org/stable/modules/generated/s
klearn.preprocessing.StandardScaler.html
from sklearn.preprocessing import StandardScaler

# price_standardized = standardScalar.fit(project_data['price'].values)
# this will rise the error
# ValueError: Expected 2D array, got 1D array instead: array=[725.05 213.03 32
9.   ... 399.   287.73   5.5 ].
# Reshape your data either using array.reshape(-1, 1)

price_scalar = StandardScaler()
price_scalar.fit(project_data['price'].values.reshape(-1,1)) # finding the mea
n and standard deviation of this data
print(f"Mean : {price_scalar.mean_[0]}, Standard deviation : {np.sqrt(price_sc
alar.var_[0])}")

# Now standardize the data with above maen and variance.
price_standardized = price_scalar.transform(project_data['price'].values.resha
pe(-1, 1))
```

```
Mean : 299.16610437500003, Standard deviation : 375.7800454521539
```

In [47]:
```
price_standardized
```

Out[47]:
```
array([[ 1.13333292],
       [-0.22921947],
       [ 0.07939191],
       ...,
       [-0.08142025],
       [ 0.26567109],
       [-0.78148403]])
```

In [48]:
```
# Vectorizing teacher_number_of_previously_posted_projects
teacher_number_of_previously_posted_projects_scalar = StandardScaler()
teacher_number_of_previously_posted_projects_scalar.fit(project_data['teacher_
number_of_previously_posted_projects'].values.reshape(-1,1)) # finding the mea
n and standard deviation of this data
print(f"Mean : {teacher_number_of_previously_posted_projects_scalar.mean_[0]},
Standard deviation : {np.sqrt(teacher_number_of_previously_posted_projects_sca
lar.var_[0])}")

# Now standardize the data with above maen and variance.
teacher_number_of_previously_posted_projects_standardized = teacher_number_of_
previously_posted_projects_scalar.transform(project_data['teacher_number_of_pr
eviously_posted_projects'].values.reshape(-1, 1))
```

```
Mean : 11.214825, Standard deviation : 27.961699254862445
```

1.5.4 Merging all the above features

- we need to merge all the numerical vectors i.e catogorical, text, numerical vectors

In [49]:
```
# Categorical
print(school_state_one_hot.shape)
print(categories_one_hot.shape)
print(sub_categories_one_hot.shape)
print(teacher_prefix_one_hot.shape)
print(grade_one_hot.shape)
print(text_bow_titles.shape)
print(text_bow.shape)
# Numerical
print(price_standardized.shape)
print(teacher_number_of_previously_posted_projects_standardized.shape)
```

```
(80000, 51)
(80000, 9)
(80000, 30)
(80000, 5)
(80000, 4)
(80000, 2740)
(80000, 14627)
(80000, 1)
(80000, 1)
```

In [50]:
```
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
# with the same hstack function we are concatinating a sparse matrix and a den
se matirx :)
X = hstack((school_state_one_hot,categories_one_hot, sub_categories_one_hot, t
eacher_prefix_one_hot,
            grade_one_hot, text_bow_titles, text_bow, price_standardized,
            teacher_number_of_previously_posted_projects_standardized))
X.shape
```

Out[50]:
```
(80000, 17468)
```

In [51]:
```
X.__class__
```

Out[51]:
```
scipy.sparse.coo.coo_matrix
```

Assignment 3: Apply KNN

1. **[Task-1] Apply KNN(brute force version) on these feature sets**
    - Set 1: categorical, numerical features + project_title(BOW) + preprocessed_essay (BOW)
    - Set 2: categorical, numerical features + project_title(TFIDF)+ preprocessed_essay (TFIDF)
    - Set 3: categorical, numerical features + project_title(AVG W2V)+ preprocessed_essay (AVG W2V)
    - Set 4: categorical, numerical features + project_title(TFIDF W2V)+ preprocessed_essay (TFIDF W2V)

2. **Hyper paramter tuning to find best K**
    - Find the best hyper parameter which results in the maximum AUC (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/receiver-operating-characteristic-curve-roc-curve-and-auc-1/) value
    - Find the best hyper paramter using k-fold cross validation (or) simple cross validation data
    - Use gridsearch-cv or randomsearch-cv or write your own for loops to do this task

3. **Representation of results**
    - You need to plot the performance of model both on train data and cross validation data for each hyper parameter, as shown in the figure
    - Once you find the best hyper parameter, you need to train your model-M using the best hyper-param. Now, find the AUC on test data and plot the ROC curve on both train and test using model-M.
    - Along with plotting ROC curve, you need to print the confusion matrix (https://www.appliedaicourse.com/course/applied-ai-course-online/lessons/confusion-matrix-tpr-fpr-fnr-tnr-1/) with predicted and original labels of test data points

4. **[Task-2]**
    - Select top 2000 features from feature Set 2 using `SelectKBest` (https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html) and then apply KNN on top of these features
    -
    ```
    from sklearn.datasets import load_digits
    from sklearn.feature_selection import SelectKBest, chi2
    X, y = load_digits(return_X_y=True)
    X.shape
    X_new = SelectKBest(chi2, k=20).fit_transform(X, y)
    X_new.shape
    ========
    output:
    (1797, 64)
    (1797, 20)
    ```
    - Repeat the steps 2 and 3 on the data matrix after feature selection

5. **Conclusion**

- You need to summarize the results at the end of the notebook, summarize it in the table format. To print out a table please refer to this prettytable library [link (http://zetcode.com/python/prettytable/)](http://zetcode.com/python/prettytable/)

Note: Data Leakage

1. There will be an issue of data-leakage if you vectorize the entire data and then split it into train/cv/test.
2. To avoid the issue of data-leakag, make sure to split your data first and then vectorize it.
3. While vectorizing your data, apply the method fit_transform() on you train data, and apply the method transform() on cv/test data.
4. For more details please go through this [link. (https://soundcloud.com/applied-ai-course/leakage-bow-and-tfidf)](https://soundcloud.com/applied-ai-course/leakage-bow-and-tfidf)

2. K Nearest Neighbor

In [60]:
```
project_data.head(3)
```

Out[60]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	Date
0	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs	CA	2016-04-27 00:27:36
1	37728	p043609	3f60494c61921b3b43ab61bdde2904df	Ms	UT	2016-04-27 00:31:25
2	74477	p189804	4a97f3a390bfe21b99cf5e2b81981c73	Mrs	CA	2016-04-27 00:46:53

In [61]:
```
y = project_data['project_is_approved'].values
X = project_data.drop(['project_is_approved'], axis=1)
X.head(1)
```

Out[61]:

	Unnamed: 0	id	teacher_id	teacher_prefix	school_state	Date
0	8393	p205479	2bf07ba08945e5d8b2a3f269b2b3cfe5	Mrs	CA	2016-04-27 00:27:36

2.1 Splitting data into Train and cross validation(or test): Stratified Sampling

In [63]:
```
# train test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, stra
tify=y)
X_train, X_cv, y_train, y_cv = train_test_split(X_train, y_train, test_size=0.
33)
```

In [64]:
```
print(X_train.shape, y_train.shape)
print(X_cv.shape, y_cv.shape)
print(X_test.shape, y_test.shape)

print("="*100)
```

```
(35912, 19) (35912,)
(17688, 19) (17688,)
(26400, 19) (26400,)
=============================================================================
=======================
```

2.2 Make Data Model Ready: encoding numerical, categorical features

In [71]:

```
# School State
vectorizer = CountVectorizer()
vectorizer.fit(X_train['school_state'].values) # fit has to happen only on tra
in data

# we use the fitted CountVectorizer to convert the text to vector
X_train_state_ohe = vectorizer.transform(X_train['school_state'].values)
X_cv_state_ohe = vectorizer.transform(X_cv['school_state'].values)
X_test_state_ohe = vectorizer.transform(X_test['school_state'].values)

print("After vectorizations")
print(X_train_state_ohe.shape, y_train.shape)
print(X_cv_state_ohe.shape, y_cv.shape)
print(X_test_state_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(35912, 51) (35912,)
(17688, 51) (17688,)
(26400, 51) (26400,)
['ak', 'al', 'ar', 'az', 'ca', 'co', 'ct', 'dc', 'de', 'fl', 'ga', 'hi', 'i
a', 'id', 'il', 'in', 'ks', 'ky', 'la', 'ma', 'md', 'me', 'mi', 'mn', 'mo',
'ms', 'mt', 'nc', 'nd', 'ne', 'nh', 'nj', 'nm', 'nv', 'ny', 'oh', 'ok', 'or',
'pa', 'ri', 'sc', 'sd', 'tn', 'tx', 'ut', 'va', 'vt', 'wa', 'wi', 'wv', 'wy']
============================================================================
========================
```

In [72]:

```
# teacher_prefix
vectorizer = CountVectorizer()
vectorizer.fit(X_train['teacher_prefix'].values) # fit has to happen only on t
rain data

# we use the fitted CountVectorizer to convert the text to vector
X_train_teacher_ohe = vectorizer.transform(X_train['teacher_prefix'].values)
X_cv_teacher_ohe = vectorizer.transform(X_cv['teacher_prefix'].values)
X_test_teacher_ohe = vectorizer.transform(X_test['teacher_prefix'].values)

print("After vectorizations")
print(X_train_teacher_ohe.shape, y_train.shape)
print(X_cv_teacher_ohe.shape, y_cv.shape)
print(X_test_teacher_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(35912, 5) (35912,)
(17688, 5) (17688,)
(26400, 5) (26400,)
['dr', 'mr', 'mrs', 'ms', 'teacher']
============================================================================
========================
```

In [73]:
```
# project_grade_category
vectorizer = CountVectorizer()
vectorizer.fit(X_train['project_grade_category'].values) # fit has to happen o
nly on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_grade_ohe = vectorizer.transform(X_train['project_grade_category'].val
ues)
X_cv_grade_ohe = vectorizer.transform(X_cv['project_grade_category'].values)
X_test_grade_ohe = vectorizer.transform(X_test['project_grade_category'].value
s)

print("After vectorizations")
print(X_train_grade_ohe.shape, y_train.shape)
print(X_cv_grade_ohe.shape, y_cv.shape)
print(X_test_grade_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(35912, 4) (35912,)
(17688, 4) (17688,)
(26400, 4) (26400,)
['grades_3_5', 'grades_6_8', 'grades_9_12', 'grades_prek_2']
==============================================================================
======================
```

In [77]:
```
# categories
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_categories'].values) # fit has to happen only on
train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_category_ohe = vectorizer.transform(X_train['clean_categories'].values
)
X_cv_category_ohe = vectorizer.transform(X_cv['clean_categories'].values)
X_test_category_ohe = vectorizer.transform(X_test['clean_categories'].values)

print("After vectorizations")
print(X_train_category_ohe.shape, y_train.shape)
print(X_cv_category_ohe.shape, y_cv.shape)
print(X_test_category_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(35912, 9) (35912,)
(17688, 9) (17688,)
(26400, 9) (26400,)
['appliedlearning', 'care_hunger', 'health_sports', 'history_civics', 'litera
cy_language', 'math_science', 'music_arts', 'specialneeds', 'warmth']
==============================================================================
======================
```

In [78]:
```
# sub categories
vectorizer = CountVectorizer()
vectorizer.fit(X_train['clean_subcategories'].values) # fit has to happen only
on train data

# we use the fitted CountVectorizer to convert the text to vector
X_train_subcategory_ohe = vectorizer.transform(X_train['clean_subcategories'].
values)
X_cv_subcategory_ohe = vectorizer.transform(X_cv['clean_subcategories'].values
)
X_test_subcategory_ohe = vectorizer.transform(X_test['clean_subcategories'].va
lues)

print("After vectorizations")
print(X_train_subcategory_ohe.shape, y_train.shape)
print(X_cv_subcategory_ohe.shape, y_cv.shape)
print(X_test_subcategory_ohe.shape, y_test.shape)
print(vectorizer.get_feature_names())
print("="*100)
```

```
After vectorizations
(35912, 30) (35912,)
(17688, 30) (17688,)
(26400, 30) (26400,)
['appliedsciences', 'care_hunger', 'charactereducation', 'civics_government',
'college_careerprep', 'communityservice', 'earlydevelopment', 'economics', 'e
nvironmentalscience', 'esl', 'extracurricular', 'financialliteracy', 'foreign
languages', 'gym_fitness', 'health_lifescience', 'health_wellness', 'history_
geography', 'literacy', 'literature_writing', 'mathematics', 'music', 'nutrit
ioneducation', 'other', 'parentinvolvement', 'performingarts', 'socialscience
s', 'specialneeds', 'teamsports', 'visualarts', 'warmth']
============================================================================
========================
```

Encoding numerical features

In [81]:
```
from sklearn.preprocessing import Normalizer
normalizer = Normalizer()
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['price'].values.reshape(-1,1))

X_train_price_norm = normalizer.transform(X_train['price'].values.reshape(-1,1
))
X_cv_price_norm = normalizer.transform(X_cv['price'].values.reshape(-1,1))
X_test_price_norm = normalizer.transform(X_test['price'].values.reshape(-1,1))

print("After vectorizations")
print(X_train_price_norm.shape, y_train.shape)
print(X_cv_price_norm.shape, y_cv.shape)
print(X_test_price_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(35912, 1) (35912,)
(17688, 1) (17688,)
(26400, 1) (26400,)
==============================================================================
======================
```

In [82]:
```
# teacher previously posted projects
normalizer = Normalizer()
# normalizer.fit(X_train['price'].values)
# this will rise an error Expected 2D array, got 1D array instead:
# array=[105.22 215.96  96.01 ... 368.98  80.53 709.67].
# Reshape your data either using
# array.reshape(-1, 1) if your data has a single feature
# array.reshape(1, -1)  if it contains a single sample.
normalizer.fit(X_train['teacher_number_of_previously_posted_projects'].values.
reshape(-1,1))

X_train_teach_prev_norm = normalizer.transform(X_train['teacher_number_of_prev
iously_posted_projects'].values.reshape(-1,1))
X_cv_teach_prev_norm = normalizer.transform(X_cv['teacher_number_of_previously
_posted_projects'].values.reshape(-1,1))
X_test_teach_prev_norm = normalizer.transform(X_test['teacher_number_of_previo
usly_posted_projects'].values.reshape(-1,1))

print("After vectorizations")
print(X_train_teach_prev_norm.shape, y_train.shape)
print(X_cv_teach_prev_norm.shape, y_cv.shape)
print(X_test_teach_prev_norm.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(35912, 1) (35912,)
(17688, 1) (17688,)
(26400, 1) (26400,)
==============================================================================
======================
```

2.3 Make Data Model Ready: encoding essay, and project_title

In [65]:
```
vectorizer = CountVectorizer(min_df=10,ngram_range=(1,4), max_features=5000)
vectorizer.fit(X_train['essay'].values) # fit has to happen only on train data
```

Out[65]:
```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='conten
t',
                lowercase=True, max_df=1.0, max_features=5000, min_df=10,
                ngram_range=(1, 4), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

In [66]:
```
# we use the fitted CountVectorizer to convert the text to vector
X_train_essay_bow = vectorizer.transform(X_train['essay'].values)
X_cv_essay_bow = vectorizer.transform(X_cv['essay'].values)
X_test_essay_bow = vectorizer.transform(X_test['essay'].values)
```

In [67]:
```
print("After vectorizations")
print(X_train_essay_bow.shape, y_train.shape)
print(X_cv_essay_bow.shape, y_cv.shape)
print(X_test_essay_bow.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(35912, 5000) (35912,)
(17688, 5000) (17688,)
(26400, 5000) (26400,)
==============================================================================
======================
```

In [68]:
```
# Preprocessing project_title
vectorizer = CountVectorizer(min_df=10,ngram_range=(1,4), max_features=5000)
vectorizer.fit(X_train['project_title'].values) # fit has to happen only on tr
ain data
```

Out[68]:
```
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='conten
t',
                lowercase=True, max_df=1.0, max_features=5000, min_df=10,
                ngram_range=(1, 4), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

In [69]:
```
# we use the fitted CountVectorizer to convert the text to vector
X_train_pj_title_bow = vectorizer.transform(X_train['project_title'].values)
X_cv_pj_title_bow = vectorizer.transform(X_cv['project_title'].values)
X_test_pj_title_bow = vectorizer.transform(X_test['project_title'].values)
```

In [70]:
```
print("After vectorizations")
print(X_train_pj_title_bow.shape, y_train.shape)
print(X_cv_pj_title_bow.shape, y_cv.shape)
print(X_test_pj_title_bow.shape, y_test.shape)
print("="*100)
```

```
After vectorizations
(35912, 4272) (35912,)
(17688, 4272) (17688,)
(26400, 4272) (26400,)
==============================================================================
======================
```

2.4 Appling KNN on different kind of featurization as mentioned in the instructions

Apply KNN on different kind of featurization as mentioned in the instructions
For Every model that you work on make sure you do the step 2 and step 3 of instructions

2.4.1 Applying KNN brute force on BOW, SET 1

In [83]:
```
# concatinating all the features
# merge two sparse matrices: https://stackoverflow.com/a/19710648/4084039
from scipy.sparse import hstack
X_tr = hstack((X_train_essay_bow, X_train_state_ohe, X_train_teacher_ohe,
               X_train_grade_ohe, X_train_price_norm, X_train_category_ohe,
               X_train_subcategory_ohe, X_train_teach_prev_norm,
               X_train_pj_title_bow)).tocsr()

X_cr = hstack((X_cv_essay_bow, X_cv_state_ohe, X_cv_teacher_ohe,
               X_cv_grade_ohe, X_cv_category_ohe, X_cv_subcategory_ohe,
               X_cv_price_norm, X_cv_teach_prev_norm, X_cv_pj_title_bow)).tocs
r()

X_te = hstack((X_test_essay_bow, X_test_state_ohe, X_test_teacher_ohe,
               X_test_grade_ohe, X_test_category_ohe, X_test_subcategory_ohe,
               X_test_price_norm, X_test_teach_prev_norm,
               X_test_pj_title_bow)).tocsr()
```

In [84]:
```
print("Final Data matrix - for set 1")
print(X_tr.shape, y_train.shape)
print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix - for set 1
(35912, 9373) (35912,)
(17688, 9373) (17688,)
(26400, 9373) (26400,)
=============================================================================
=======================
```

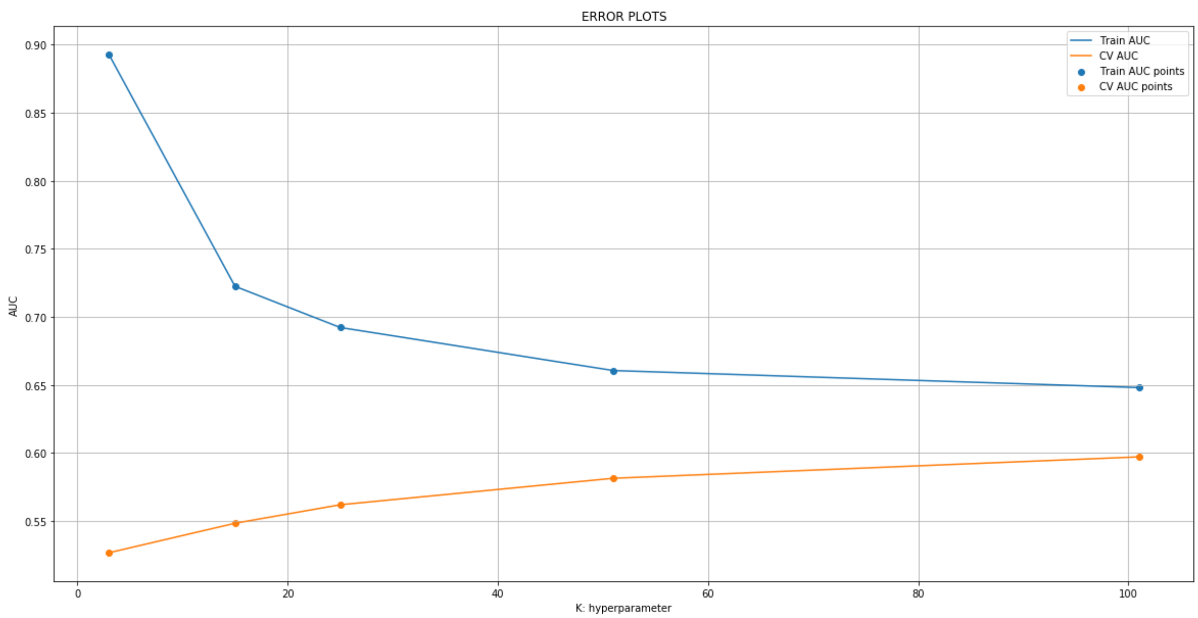
In [85]:
```
# Since there was memory errors while trying with RandomizedSearchCV, trying t
he for loop approach
def batch_predict(clf, data):
    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs

    y_data_pred = []
    tr_loop = data.shape[0] - data.shape[0]%1000
    # consider you X_tr shape is 49041, then your tr_loop will be 49041 - 4904
1%1000 = 49000
    # in this for loop we will iterate unti the last 1000 multiplier
    for i in range(0, tr_loop, 1000):
        y_data_pred.extend(clf.predict_proba(data[i:i+1000])[:,1])
    # we will be predicting for the last data points
    if data.shape[0]%1000 !=0:
        y_data_pred.extend(clf.predict_proba(data[tr_loop:])[:,1])

    return y_data_pred
```

In [86]:
```python
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import roc_auc_score
```

In [87]:
```python
train_auc = []
cv_auc = []
K = [3, 15, 25, 51, 101]
for i in tqdm(K):
    neigh = KNeighborsClassifier(n_neighbors=i, n_jobs=-1)
    neigh.fit(X_tr, y_train)

    y_train_pred = batch_predict(neigh, X_tr)
    y_cv_pred = batch_predict(neigh, X_cr)

    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
 20%|████████████████▍
| 1/5 [01:36<06:24, 96.09s/it]
 40%|████████████████████████████████▊
| 2/5 [03:18<04:53, 97.99s/it]
 60%|█████████████████████████████████████████████████▏
| 3/5 [05:01<03:18, 99.38s/it]
 80%|█████████████████████████████████████████████████████████████████▌
| 4/5 [06:44<01:40, 100.46s/it]
100%|██████████████████████████████████████████████████████████████████████
███████████| 5/5 [08:27<00:00, 101.43s/it]
```

In [88]:
```
plt.figure(figsize=(20,10))
plt.plot(K, train_auc, label='Train AUC')
plt.plot(K, cv_auc, label='CV AUC')

plt.scatter(K, train_auc, label='Train AUC points')
plt.scatter(K, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```

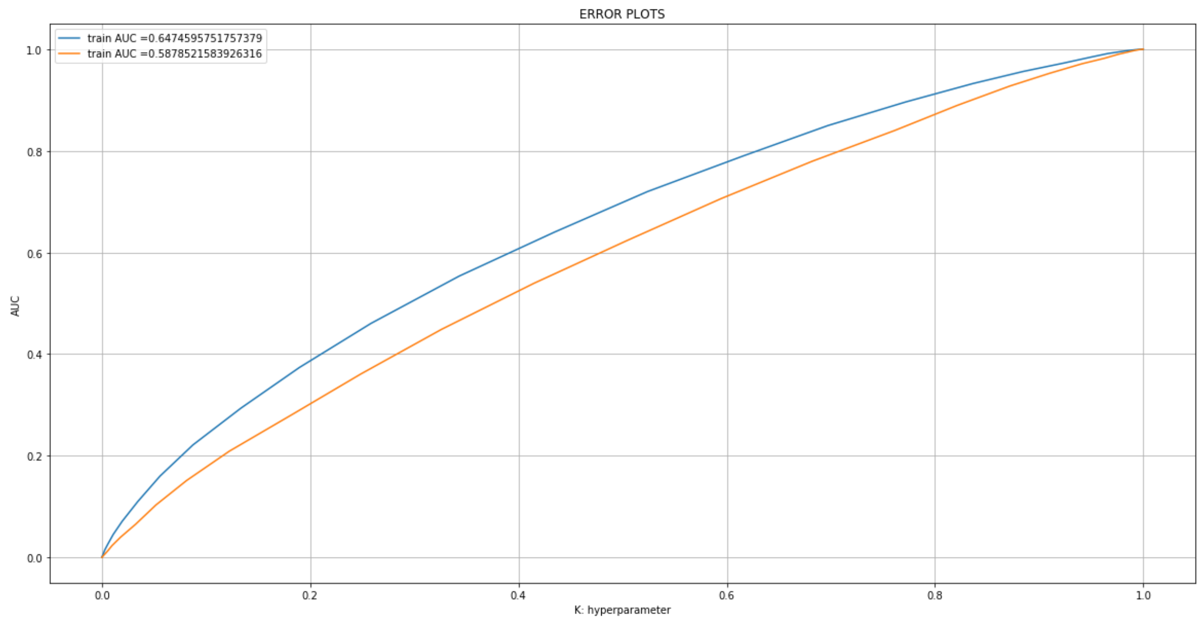


In [89]:
```
# from the above error plot, we have max AUC for CV with least difference betw
een Train and CV AUCs at K=105
best_k = 105
```

Testing the performance on test data, plotting ROC Curves

In [90]:
```
from sklearn.metrics import roc_curve, auc


neigh = KNeighborsClassifier(n_neighbors=best_k, n_jobs=-1)
neigh.fit(X_tr, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estim
ates of the positive class
# not the predicted outputs

y_train_pred = batch_predict(neigh, X_tr)
y_test_pred = batch_predict(neigh, X_te)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
```

In [91]:
```
plt.figure(figsize=(20,10))
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tp
r)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



The Blue plot in the Graph is Train AUC and Orange Plot is Test AUC. Initially the label was accidentally copy pasted as Train AUC for both. Since redrawing plot would require to re run entire ipynb and it takes up a lot of time. so documenting the information as a markdown.

In [92]:
```
# we are writing our own function for predict, with defined thresould
# we will pick a threshold that will give the least fpr
def find_best_threshold(threshould, fpr, tpr):
    t = threshould[np.argmax(tpr*(1-fpr))]
    # (tpr*(1-fpr)) will be maximum if your fpr is very low and tpr is very hi
gh
    print("the maximum value of tpr*(1-fpr)", max(tpr*(1-fpr)), "for threshol
d", np.round(t,3))
    return t

def predict_with_best_t(proba, threshould):
    predictions = []
    for i in proba:
        if i>=threshould:
            predictions.append(1)
        else:
            predictions.append(0)
    return predictions
```

In [93]:
```
print("="*100)
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(tr_thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best_t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_t)))
```

```
============================================================================
========================
the maximum value of tpr*(1-fpr) 0.36356411719545345 for threshold 0.838
Train confusion matrix
[[ 3599  1880]
 [13589 16844]]
Test confusion matrix
[[ 2342  1660]
 [10321 12077]]
```

2.4.2 Applying KNN brute force on TFIDF, SET 2

In [96]:
```
# preprocessing TFIDF of Text Essays and Project Titles
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(min_df=10, ngram_range=(1,4), max_features=5000)
vectorizer.fit(X_train["essay"].values)
X_train_essay_tfidf = vectorizer.transform(X_train['essay'].values)
X_cv_essay_tfidf = vectorizer.transform(X_cv['essay'].values)
X_test_essay_tfidf = vectorizer.transform(X_test['essay'].values)

print("Shape of Datamatrix after TFIDF Vectorization")
print(X_train_essay_tfidf.shape, y_train.shape)
print(X_cv_essay_tfidf.shape, y_cv.shape)
print(X_test_essay_tfidf.shape, y_test.shape)
print("="*100)
```

```
Shape of Datamatrix after TFIDF Vectorization
(35912, 5000) (35912,)
(17688, 5000) (17688,)
(26400, 5000) (26400,)
============================================================================
========================
```

In [97]:
```
# Similarly you can vectorize for title also
vectorizer_titles = TfidfVectorizer(min_df=10, ngram_range=(1,4), max_features
=5000)
vectorizer_titles.fit(X_train["project_title"])

X_train_pj_title_tfidf = vectorizer.transform(X_train['project_title'].values)
X_cv_pj_title_tfidf = vectorizer.transform(X_cv['project_title'].values)
X_test_pj_title_tfidf = vectorizer.transform(X_test['project_title'].values)

print("Shape of Datamatrix after TFIDF Vectorization")
print(X_train_pj_title_tfidf.shape, y_train.shape)
print(X_cv_pj_title_tfidf.shape, y_cv.shape)
print(X_test_pj_title_tfidf.shape, y_test.shape)
print("="*100)
```

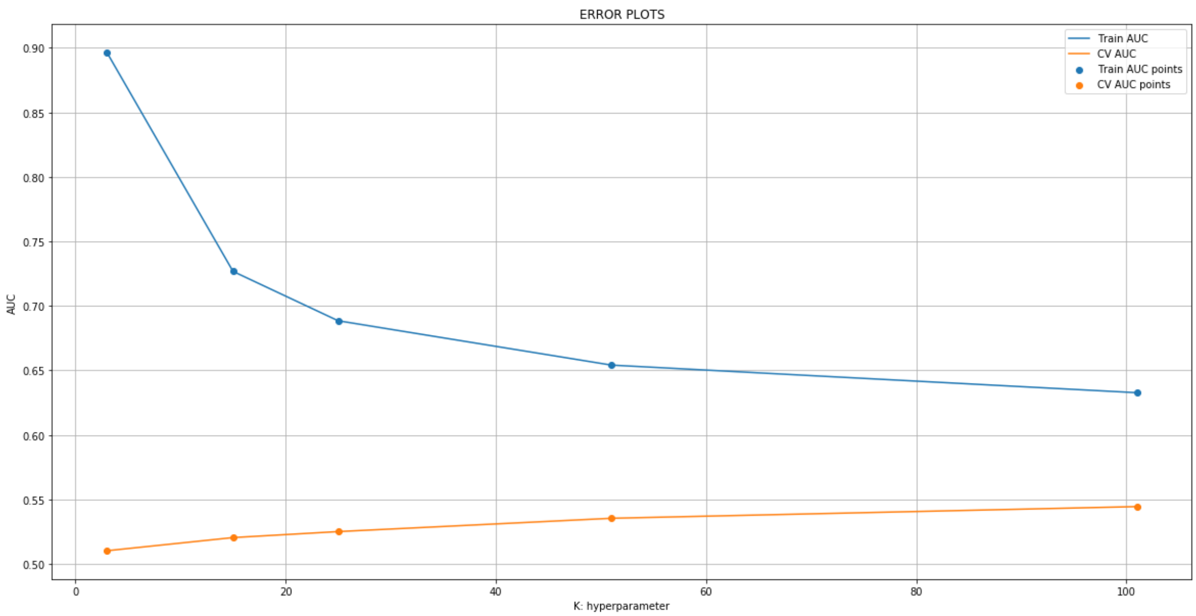
```
Shape of Datamatrix after TFIDF Vectorization
(35912, 5000) (35912,)
(17688, 5000) (17688,)
(26400, 5000) (26400,)
=============================================================================
=======================
```

In [98]:
```
# Concatinating all the features for Set 2

X_tr = hstack((X_train_essay_tfidf, X_train_state_ohe, X_train_teacher_ohe,
               X_train_grade_ohe, X_train_price_norm, X_train_category_ohe,
               X_train_subcategory_ohe, X_train_teach_prev_norm,
               X_train_pj_title_tfidf)).tocsr()

X_cr = hstack((X_cv_essay_tfidf, X_cv_state_ohe, X_cv_teacher_ohe,
               X_cv_grade_ohe, X_cv_category_ohe, X_cv_subcategory_ohe,
               X_cv_price_norm, X_cv_teach_prev_norm, X_cv_pj_title_tfidf)).to
csr()

X_te = hstack((X_test_essay_tfidf, X_test_state_ohe, X_test_teacher_ohe,
               X_test_grade_ohe, X_test_category_ohe, X_test_subcategory_ohe,
               X_test_price_norm, X_test_teach_prev_norm,
               X_test_pj_title_tfidf)).tocsr()
```

In [99]:
```
train_auc = []
cv_auc = []
K = [3, 15, 25, 51, 101]
for i in tqdm(K):
    neigh = KNeighborsClassifier(n_neighbors=i, n_jobs=-1)
    neigh.fit(X_tr, y_train)

    y_train_pred = batch_predict(neigh, X_tr)
    y_cv_pred = batch_predict(neigh, X_cr)

    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
```

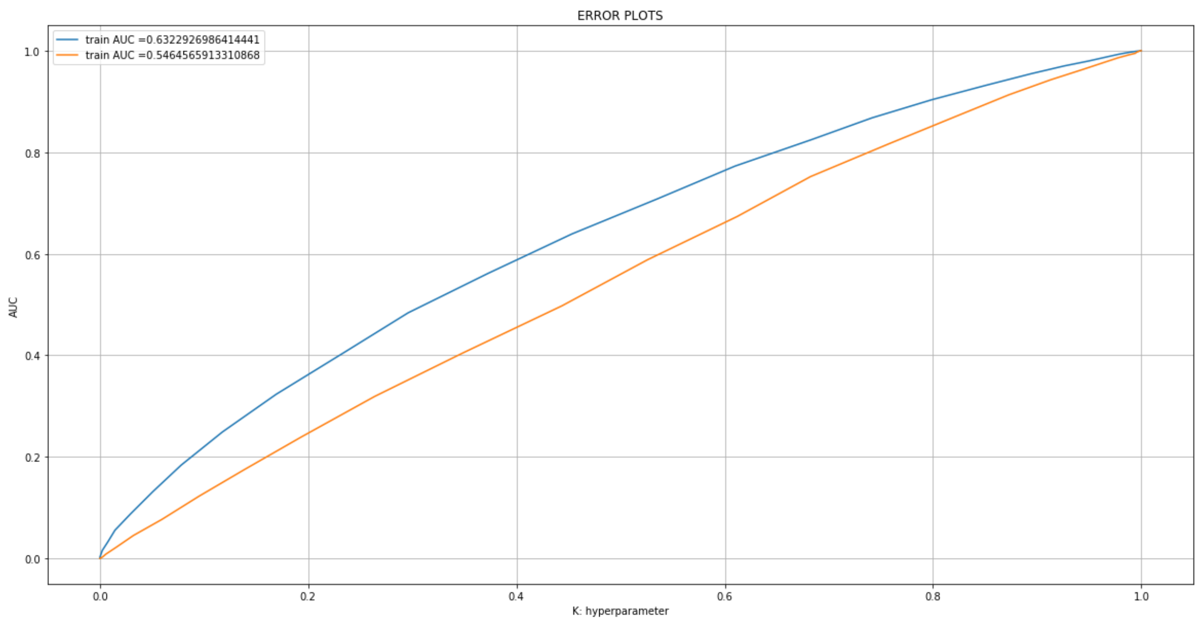
```
  0%|
| 0/5 [00:00<?, ?it/s]
 20%|█████████████████
| 1/5 [01:35<06:22, 95.53s/it]
 40%|█████████████████████████████████
| 2/5 [03:18<04:53, 97.89s/it]
 60%|██████████████████████████████████████████████████
| 3/5 [05:02<03:18, 99.49s/it]
 80%|██████████████████████████████████████████████████████████████████
| 4/5 [06:45<01:40, 100.61s/it]
100%|████████████████████████████████████████████████████████████████████████
███████████| 5/5 [08:32<00:00, 102.45s/it]
```

In [100]:
```
plt.figure(figsize=(20,10))
plt.plot(K, train_auc, label='Train AUC')
plt.plot(K, cv_auc, label='CV AUC')

plt.scatter(K, train_auc, label='Train AUC points')
plt.scatter(K, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



In [101]:
```
best_k = 105
# Testing the performance of the model on test data, plotting ROC curves
neigh = KNeighborsClassifier(n_neighbors=best_k, n_jobs=-1)
neigh.fit(X_tr, y_train)

y_train_pred = batch_predict(neigh, X_tr)
y_test_pred = batch_predict(neigh, X_te)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
```

In [102]:
```
plt.figure(figsize=(20,10))
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tp
r)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



The Blue plot in the Graph is Train AUC and Orange Plot is Test AUC. Initially the label was accidentally copy pasted as Train AUC for both. Since redrawing plot would require to re run entire ipynb and it takes up a lot of time. so documenting the information as a markdown.

In [103]:
```
# Confusion matrix
print("="*100)
from sklearn.metrics import confusion_matrix
best_t = find_best_threshold(tr_thresholds, train_fpr, train_tpr)
print("Train confusion matrix")
print(confusion_matrix(y_train, predict_with_best_t(y_train_pred, best_t)))
print("Test confusion matrix")
print(confusion_matrix(y_test, predict_with_best_t(y_test_pred, best_t)))
```

```
=============================================================================
=======================
the maximum value of tpr*(1-fpr) 0.35208438007015214 for threshold 0.857
Train confusion matrix
[[ 3437  2042]
 [13352 17081]]
Test confusion matrix
[[ 1898  2104]
 [ 9227 13171]]
```

2.4.3 Applying KNN brute force on AVG W2V, SET 3

In [104]:
```
# Please write all the code with proper documentation
# make sure you have the glove_vectors file
with open('glove_vectors', 'rb') as f:
    model = pickle.load(f)
    glove_words =  set(model.keys())
```

In [105]:
```
# average Word2Vec
# compute average word2vec for each review.
avg_w2v_vectors_train = []; # the avg-w2v for each sentence/review is stored i
n this list
for sentence in tqdm(X_train['essay'].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_train.append(vector)

print(len(avg_w2v_vectors_train))
print(len(avg_w2v_vectors_train[0]))
print(avg_w2v_vectors_train[0])
```

```
  0%|
| 0/35912 [00:00<?, ?it/s]
  1%|
| 304/35912 [00:00<00:11, 3017.95it/s]
  2%|
| 588/35912 [00:00<00:11, 2955.51it/s]
  2%|
| 876/35912 [00:00<00:11, 2925.92it/s]
  3%|
| 1185/35912 [00:00<00:11, 2967.03it/s]
  4%|
| 1478/35912 [00:00<00:11, 2949.30it/s]
  5%|
| 1786/35912 [00:00<00:11, 2980.99it/s]
  6%|
| 2069/35912 [00:00<00:11, 2927.37it/s]
  7%|
| 2380/35912 [00:00<00:11, 2973.47it/s]
  7%|
| 2661/35912 [00:00<00:11, 2845.31it/s]
  8%|
| 2975/35912 [00:01<00:11, 2921.77it/s]
  9%|
| 3289/35912 [00:01<00:10, 2977.64it/s]
 10%|
| 3589/35912 [00:01<00:10, 2977.96it/s]
 11%|
| 3884/35912 [00:01<00:10, 2954.11it/s]
 12%|
| 4178/35912 [00:01<00:11, 2784.89it/s]
 12%|
| 4457/35912 [00:01<00:11, 2763.87it/s]
 13%|
| 4765/35912 [00:01<00:10, 2845.89it/s]
 14%|
| 5085/35912 [00:01<00:10, 2937.69it/s]
 15%|
| 5410/35912 [00:01<00:10, 3018.73it/s]
 16%|
| 5714/35912 [00:01<00:10, 2825.43it/s]
 17%|
| 6007/35912 [00:02<00:10, 2849.92it/s]
 18%|
| 6303/35912 [00:02<00:10, 2875.95it/s]
 18%|
| 6593/35912 [00:02<00:10, 2762.07it/s]
 19%|
| 6893/35912 [00:02<00:10, 2823.56it/s]
 20%|
| 7196/35912 [00:02<00:09, 2876.47it/s]
 21%|
| 7501/35912 [00:02<00:09, 2920.28it/s]
 22%|
| 7800/35912 [00:02<00:09, 2934.53it/s]
 23%|
| 8095/35912 [00:02<00:09, 2932.74it/s]
 23%|
```

```
| 8402/35912 [00:02<00:09, 2966.33it/s]
 24%|
| 8700/35912 [00:02<00:09, 2869.99it/s]
 25%|
| 8989/35912 [00:03<00:09, 2762.99it/s]
 26%|
| 9267/35912 [00:03<00:10, 2659.27it/s]
 27%|
| 9560/35912 [00:03<00:09, 2729.50it/s]
 27%|
| 9835/35912 [00:03<00:09, 2705.56it/s]
 28%|
| 10110/35912 [00:03<00:09, 2712.86it/s]
 29%|
| 10383/35912 [00:03<00:09, 2704.03it/s]
 30%|
| 10673/35912 [00:03<00:09, 2754.24it/s]
 31%|
| 10967/35912 [00:03<00:08, 2801.59it/s]
 31%|
| 11273/35912 [00:03<00:08, 2868.50it/s]
 32%|
| 11569/35912 [00:04<00:08, 2889.17it/s]
 33%|
| 11885/35912 [00:04<00:08, 2959.32it/s]
 34%|
| 12198/35912 [00:04<00:07, 3002.21it/s]
 35%|
| 12499/35912 [00:04<00:08, 2821.41it/s]
 36%|
| 12784/35912 [00:04<00:08, 2823.79it/s]
 36%|
| 13079/35912 [00:04<00:07, 2854.44it/s]
 37%|
| 13366/35912 [00:04<00:08, 2746.65it/s]
 38%|
| 13665/35912 [00:04<00:07, 2809.60it/s]
 39%|
| 13968/35912 [00:04<00:07, 2866.33it/s]
 40%|
| 14282/35912 [00:04<00:07, 2937.25it/s]
 41%|
| 14578/35912 [00:05<00:07, 2757.50it/s]
 41%|
| 14870/35912 [00:05<00:07, 2798.42it/s]
 42%|
| 15153/35912 [00:05<00:07, 2752.80it/s]
 43%|
| 15466/35912 [00:05<00:07, 2850.36it/s]
 44%|
| 15754/35912 [00:05<00:07, 2747.13it/s]
 45%|
| 16032/35912 [00:05<00:07, 2734.73it/s]
 45%|
| 16308/35912 [00:05<00:07, 2680.64it/s]
 46%|
| 16604/35912 [00:05<00:07, 2753.13it/s]
```

```
47%|
| 16881/35912 [00:05<00:06, 2735.91it/s]
48%|
| 17183/35912 [00:06<00:06, 2809.61it/s]
49%|
| 17480/35912 [00:06<00:06, 2849.87it/s]
49%|
| 17766/35912 [00:06<00:06, 2740.57it/s]
50%|
| 18086/35912 [00:06<00:06, 2858.33it/s]
51%|
| 18375/35912 [00:06<00:06, 2771.22it/s]
52%|
| 18661/35912 [00:06<00:06, 2791.28it/s]
53%|
| 18968/35912 [00:06<00:05, 2863.58it/s]
54%|
| 19275/35912 [00:06<00:05, 2916.44it/s]
54%|
| 19569/35912 [00:06<00:05, 2800.71it/s]
55%|
| 19872/35912 [00:06<00:05, 2859.83it/s]
56%|
| 20176/35912 [00:07<00:05, 2905.51it/s]
57%|
| 20484/35912 [00:07<00:05, 2949.53it/s]
58%|
| 20781/35912 [00:07<00:05, 2831.44it/s]
59%|
| 21066/35912 [00:07<00:05, 2710.00it/s]
59%|
| 21367/35912 [00:07<00:05, 2787.85it/s]
60%|
| 21661/35912 [00:07<00:05, 2825.85it/s]
61%|
| 21983/35912 [00:07<00:04, 2927.74it/s]
62%|
| 22298/35912 [00:07<00:04, 2984.84it/s]
63%|
| 22599/35912 [00:07<00:04, 2866.72it/s]
64%|
| 22896/35912 [00:08<00:04, 2890.75it/s]
65%|
| 23197/35912 [00:08<00:04, 2919.31it/s]
65%|
| 23491/35912 [00:08<00:04, 2910.47it/s]
66%|
| 23783/35912 [00:08<00:04, 2906.98it/s]
67%|
| 24075/35912 [00:08<00:04, 2870.36it/s]
68%|
| 24363/35912 [00:08<00:04, 2784.04it/s]
69%|
| 24661/35912 [00:08<00:03, 2834.15it/s]
70%|
| 24960/35912 [00:08<00:03, 2873.11it/s]
70%|
```

```
| 25271/35912 [00:08<00:03, 2934.22it/s]
71%|
| 25580/35912 [00:08<00:03, 2973.00it/s]
72%|
| 25879/35912 [00:09<00:03, 2804.78it/s]
73%|
| 26187/35912 [00:09<00:03, 2876.15it/s]
74%|
| 26477/35912 [00:09<00:03, 2778.03it/s]
75%|
| 26758/35912 [00:09<00:03, 2757.00it/s]
75%|
| 27061/35912 [00:09<00:03, 2827.80it/s]
76%|
| 27370/35912 [00:09<00:02, 2895.70it/s]
77%|
| 27676/35912 [00:09<00:02, 2936.92it/s]
78%|
| 27976/35912 [00:09<00:02, 2949.19it/s]
79%|
| 28272/35912 [00:09<00:02, 2919.85it/s]
80%|
| 28565/35912 [00:10<00:02, 2873.78it/s]
80%|
| 28854/35912 [00:10<00:02, 2757.54it/s]
81%|
| 29132/35912 [00:10<00:02, 2733.89it/s]
82%|
| 29420/35912 [00:10<00:02, 2770.28it/s]
83%|
| 29709/35912 [00:10<00:02, 2799.16it/s]
84%|
| 30007/35912 [00:10<00:02, 2845.12it/s]
84%|
| 30293/35912 [00:10<00:02, 2729.69it/s]
85%|
| 30568/35912 [00:10<00:01, 2729.79it/s]
86%|
| 30850/35912 [00:10<00:01, 2750.35it/s]
87%|
| 31149/35912 [00:10<00:01, 2812.31it/s]
88%|
| 31432/35912 [00:11<00:01, 2762.20it/s]
88%|
| 31710/35912 [00:11<00:01, 2681.78it/s]
89%|
| 32000/35912 [00:11<00:01, 2738.05it/s]
90%|
| 32275/35912 [00:11<00:01, 2679.81it/s]
91%|
| 32544/35912 [00:11<00:01, 2562.55it/s]
91%|
| 32842/35912 [00:11<00:01, 2669.72it/s]
92%|
| 33148/35912 [00:11<00:00, 2770.33it/s]
93%|
| 33428/35912 [00:11<00:00, 2709.01it/s]
```

```
 94%|█████████▍| 33702/35912 [00:11<00:00, 2596.96it/s]
 95%|█████████▍| 33988/35912 [00:12<00:00, 2665.19it/s]
 96%|█████████▌| 34297/35912 [00:12<00:00, 2774.38it/s]
 96%|█████████▋| 34578/35912 [00:12<00:00, 2675.98it/s]
 97%|█████████▋| 34849/35912 [00:12<00:00, 2610.75it/s]
 98%|█████████▊| 35144/35912 [00:12<00:00, 2698.63it/s]
 99%|█████████▊| 35434/35912 [00:12<00:00, 2750.31it/s]
100%|██████████| 35912/35912 [00:12<00:00, 2827.49it/s]
```

```
35912
300
[-3.75512795e-02 -6.69488513e-02 -6.70803363e-02 -1.90723491e-01
  4.33536981e-02 -4.46185099e-02 -3.70475594e+00  1.76297780e-01
  6.31845660e-03 -2.25834307e-01  1.01603749e-01  6.71636127e-03
  3.48734495e-02 -8.40747873e-02 -1.04109690e-01 -6.75093146e-02
 -9.43774835e-02 -7.41148665e-02  1.05182125e-01  9.59291340e-02
  2.52670566e-02 -4.15442519e-02 -6.39903320e-02  9.43120042e-02
 -5.90282906e-02 -7.06461320e-02  6.07404236e-02 -9.37109480e-02
 -7.13501285e-02 -9.25726127e-02 -2.86350256e-01 -4.88240014e-02
  6.97527967e-02  3.46846363e-02 -7.01084848e-02 -7.31722137e-02
 -5.38548307e-02 -1.13977837e-01  1.75220575e-02 -7.33068595e-02
 -6.97333679e-02  1.49121928e-01 -5.89289241e-03 -2.05862483e-01
 -5.39752434e-02 -8.60852127e-02  7.63792505e-02 -1.01604432e-01
 -5.63580118e-02 -1.25356231e-02  4.37574858e-03  4.46424500e-02
 -3.00707406e-02 -4.03732665e-02  6.80637217e-02 -9.11545091e-02
  8.31546783e-02 -2.76295835e-02 -3.92472906e-02  8.98411288e-02
 -3.75046005e-02 -4.32509764e-02  7.75953461e-02 -5.26954420e-02
  3.56176792e-03  1.19512247e-01  5.98629509e-02  8.99909561e-02
  2.07070452e-01 -1.77531633e-01 -1.46071150e-01  3.66391651e-02
 -2.79509458e-02 -7.51180635e-02 -3.10207371e-02 -2.23594016e-01
  1.25378812e-01  3.30216623e-02  7.55462792e-02 -1.01349446e-01
  3.69506439e-02 -4.14687118e-01 -6.68423505e-02 -1.28211963e-01
 -2.76120401e-02  5.80354334e-02  8.18738108e-02 -9.31786991e-02
  9.92280844e-02 -5.41338708e-02  5.15280594e-02 -5.00081396e-02
 -3.32537236e-02  5.64134986e-02  2.91459538e-02 -2.82199116e-01
 -2.51290434e+00 -6.20086571e-02  1.23330066e-01  1.13975784e-01
 -1.09734866e-01  5.42183345e-02  2.74902544e-01 -2.18736425e-02
 -9.30018113e-03 -6.85015759e-03  7.03599340e-02 -1.39569992e-01
  1.52875708e-03 -2.32907156e-02 -6.03078500e-02  6.99941005e-03
  5.77007557e-02  1.85541224e-01 -7.88257231e-02  7.99913747e-02
 -1.00275681e-01 -8.50787684e-03  1.17561448e-01  5.42727670e-02
 -8.98184340e-02  2.21527170e-02  4.74494210e-02 -1.85058015e-01
  3.79218074e-02 -1.81355519e-02  4.37706354e-02 -5.77079245e-02
  7.01073726e-03  1.13459167e-01 -1.51163398e-02  5.51013995e-02
 -8.17969623e-02 -1.15731642e-01  8.39074829e-02 -3.59646696e-02
  1.03642450e-01 -3.99207349e-02  2.39044960e-01  3.59695305e-01
  1.31053569e-01  4.37903283e-02  8.22113660e-02  4.06554434e-03
 -6.46671292e-03 -3.38558616e-02  9.73816792e-02 -7.05552262e-02
  3.21097126e-01  1.03204479e-01 -7.17701604e-03 -5.84411000e-02
  3.94897877e-02 -9.15246811e-02  1.07563754e-01 -7.09258915e-02
  2.14623821e-02  4.15949281e-02 -1.46675448e-01 -2.90320335e-02
  8.26662104e-02 -8.53794514e-02 -3.91979717e-02 -7.01792995e-02
 -2.39609097e-02  3.54544764e-02 -7.54533198e-02  1.27745723e-01
  1.24336272e-01 -1.03204526e-01 -3.97040623e-02 -2.38604896e-02
 -8.77995146e-02 -1.40369784e-01 -1.14014311e-01  1.91901149e-01
 -1.04500453e-01 -2.22308863e-02 -8.49270698e-02 -7.62908456e-02
  1.19993331e-01  1.57618745e-01 -4.11730561e-02 -6.75506980e-02
  1.61752132e-02 -1.13839944e-01 -2.31425892e-02  2.74699910e-02
  1.13403331e-01  3.36503462e-03 -1.08718588e-02 -7.37779118e-02
 -9.34916019e-03  1.01493253e-02 -1.11434184e-02 -1.27577170e-01
  6.30833887e-02  6.94061448e-02  9.47414604e-02 -1.78464753e-02
  1.54618331e-01 -6.59983879e-02 -5.03701236e-02  5.30649009e-02
 -4.28139481e-02  8.20074540e-02  3.36096502e-02 -1.35731005e-01
  2.29840204e-01 -3.82775307e-02  8.40885778e-02 -3.06141199e-02
 -1.07436777e-01 -1.43445985e-01 -6.16349788e-02 -9.09813509e-02
 -9.85752830e-02 -7.56855292e-02 -1.74586554e-02  6.06722302e-02
```

```
 -9.78945961e-02 -1.13928922e-01 -4.61330797e-02 -1.80887821e-02
 -2.60652231e+00  9.17834104e-02 -6.90932764e-02  2.74259265e-02
  1.33784651e-02 -1.22021278e-01  7.16237646e-02  7.87142259e-02
 -1.79656509e-02 -2.52493118e-02 -1.09611346e-01  1.86658514e-01
 -4.44758121e-02 -3.36083245e-02 -1.01186310e-01  3.02425453e-02
 -1.13465377e-01  3.82199332e-02 -2.19400804e-01  1.19033893e-01
 -1.68650943e-02 -3.45514575e-02 -3.26362642e-02 -4.12791164e-02
 -5.13733425e-02  5.39199042e-02  1.51183019e-04 -5.22306712e-02
  7.28870057e-02 -5.81570783e-02  1.01209083e-01  7.96570038e-03
  7.70483802e-02 -5.84984816e-02  4.63630439e-02 -1.08076597e-02
  1.18586374e-01 -6.39099319e-02 -3.24825458e-02 -5.67872764e-02
  8.10332840e-02 -8.76950491e-02 -2.64661931e-01 -3.06185264e-02
  1.43698059e-01  2.25140673e-02 -3.73186382e-02 -8.39512039e-02
 -2.13309127e-01  5.59564858e-02  5.24485278e-02  6.97427901e-02
  6.31198585e-02  1.07468608e-02 -3.39265542e-02 -2.89662286e-02
  3.00834358e-01 -3.05820868e-02  2.70612792e-02  1.47800315e-01
  1.00594321e-02  1.29999731e-01  4.63337080e-03  6.27311689e-02
 -2.71931377e-02 -3.35124858e-02 -2.49807858e-03 -9.85138981e-02
 -8.56981675e-02 -7.69237292e-02 -1.47365100e-02  1.20339675e-02
 -4.93174528e-02  3.68590708e-02  1.01812524e-01  1.16092321e-02]
```

In [106]:
```
avg_w2v_vectors_cv = []; # the avg-w2v for each sentence/review is stored in t
his list
for sentence in tqdm(X_cv['essay'].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_cv.append(vector)
```

```
  0%|
| 0/17688 [00:00<?, ?it/s]
  2%|
| 301/17688 [00:00<00:05, 2988.19it/s]
  3%|
| 601/17688 [00:00<00:05, 2985.19it/s]
  5%|
| 907/17688 [00:00<00:05, 3000.78it/s]
  7%|
| 1212/17688 [00:00<00:05, 3008.86it/s]
  9%|
| 1515/17688 [00:00<00:05, 3008.61it/s]
 10%|
| 1813/17688 [00:00<00:05, 2993.36it/s]
 12%|
| 2120/17688 [00:00<00:05, 3009.47it/s]
 14%|
| 2413/17688 [00:00<00:05, 2978.52it/s]
 15%|
| 2717/17688 [00:00<00:05, 2990.25it/s]
 17%|
| 3020/17688 [00:01<00:04, 2995.56it/s]
 19%|
| 3331/17688 [00:01<00:04, 3022.55it/s]
 21%|
| 3636/17688 [00:01<00:04, 3024.14it/s]
 22%|
| 3942/17688 [00:01<00:04, 3028.23it/s]
 24%|
| 4255/17688 [00:01<00:04, 3051.53it/s]
 26%|
| 4565/17688 [00:01<00:04, 3059.28it/s]
 28%|
| 4870/17688 [00:01<00:04, 3049.79it/s]
 29%|
| 5175/17688 [00:01<00:04, 3007.27it/s]
 31%|
| 5476/17688 [00:01<00:04, 2914.56it/s]
 33%|
| 5769/17688 [00:01<00:04, 2912.81it/s]
 34%|
| 6077/17688 [00:02<00:03, 2954.80it/s]
 36%|
| 6387/17688 [00:02<00:03, 2990.58it/s]
 38%|
| 6699/17688 [00:02<00:03, 3021.82it/s]
 40%|
| 7018/17688 [00:02<00:03, 3063.93it/s]
 41%|
| 7332/17688 [00:02<00:03, 3079.73it/s]
 43%|
| 7641/17688 [00:02<00:03, 2903.14it/s]
 45%|
| 7947/17688 [00:02<00:03, 2942.26it/s]
 47%|
| 8259/17688 [00:02<00:03, 2987.14it/s]
 48%|
```

```
| 8560/17688 [00:02<00:03, 2987.45it/s]
 50%|██████████████████████████████████████
| 8874/17688 [00:02<00:02, 3025.23it/s]
 52%|███████████████████████████████████████
| 9178/17688 [00:03<00:02, 3023.04it/s]
 54%|████████████████████████████████████████
| 9488/17688 [00:03<00:02, 3039.18it/s]
 55%|█████████████████████████████████████████
| 9798/17688 [00:03<00:02, 3050.58it/s]
 57%|██████████████████████████████████████████
| 10115/17688 [00:03<00:02, 3078.88it/s]
 59%|███████████████████████████████████████████
| 10426/17688 [00:03<00:02, 3081.44it/s]
 61%|█████████████████████████████████████████████
| 10737/17688 [00:03<00:02, 3083.24it/s]
 62%|██████████████████████████████████████████████
| 11049/17688 [00:03<00:02, 3087.47it/s]
 64%|███████████████████████████████████████████████
| 11363/17688 [00:03<00:02, 3096.33it/s]
 66%|█████████████████████████████████████████████████
| 11673/17688 [00:03<00:01, 3063.25it/s]
 68%|██████████████████████████████████████████████████
| 11980/17688 [00:03<00:01, 3058.58it/s]
 70%|███████████████████████████████████████████████████
| 12304/17688 [00:04<00:01, 3104.30it/s]
 71%|█████████████████████████████████████████████████████
| 12620/17688 [00:04<00:01, 3114.06it/s]
 73%|██████████████████████████████████████████████████████
| 12932/17688 [00:04<00:01, 3099.80it/s]
 75%|███████████████████████████████████████████████████████
| 13255/17688 [00:04<00:01, 3131.07it/s]
 77%|█████████████████████████████████████████████████████████
| 13569/17688 [00:04<00:01, 3117.62it/s]
 78%|██████████████████████████████████████████████████████████
| 13881/17688 [00:04<00:01, 3102.26it/s]
 80%|███████████████████████████████████████████████████████████
| 14192/17688 [00:04<00:01, 3070.35it/s]
 82%|█████████████████████████████████████████████████████████████
| 14504/17688 [00:04<00:01, 3078.41it/s]
 84%|██████████████████████████████████████████████████████████████
| 14823/17688 [00:04<00:00, 3104.41it/s]
 86%|███████████████████████████████████████████████████████████████
| 15134/17688 [00:04<00:00, 3062.75it/s]
 87%|████████████████████████████████████████████████████████████████
| 15441/17688 [00:05<00:00, 3058.26it/s]
 89%|██████████████████████████████████████████████████████████████████
| 15754/17688 [00:05<00:00, 3072.81it/s]
 91%|███████████████████████████████████████████████████████████████████
| 16070/17688 [00:05<00:00, 3091.81it/s]
 93%|████████████████████████████████████████████████████████████████████
| 16380/17688 [00:05<00:00, 3024.42it/s]
 94%|██████████████████████████████████████████████████████████████████████
| 16683/17688 [00:05<00:00, 3001.59it/s]
 96%|███████████████████████████████████████████████████████████████████████
| 16998/17688 [00:05<00:00, 3038.18it/s]
 98%|████████████████████████████████████████████████████████████████████████
█  | 17317/17688 [00:05<00:00, 3075.67it/s]
```

```
100%|██████████████████████████████████████████████████████████████████
██| 17688/17688 [00:05<00:00, 3041.55it/s]
```

In [107]:
```
avg_w2v_vectors_test = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_test['essay'].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_test.append(vector)
```

```
  0%|
| 0/26400 [00:00<?, ?it/s]
  1%|
| 307/26400 [00:00<00:08, 3047.72it/s]
  2%|
| 581/26400 [00:00<00:08, 2941.45it/s]
  3%|
| 880/26400 [00:00<00:08, 2949.46it/s]
  4%|
| 1176/26400 [00:00<00:08, 2946.17it/s]
  6%|
| 1476/26400 [00:00<00:08, 2955.71it/s]
  7%|
| 1780/26400 [00:00<00:08, 2974.11it/s]
  8%|
| 2071/26400 [00:00<00:08, 2948.03it/s]
  9%|
| 2370/26400 [00:00<00:08, 2954.08it/s]
 10%|
| 2680/26400 [00:00<00:07, 2990.06it/s]
 11%|
| 2977/26400 [00:01<00:07, 2977.46it/s]
 12%|
| 3285/26400 [00:01<00:07, 3001.08it/s]
 14%|
| 3590/26400 [00:01<00:07, 3009.07it/s]
 15%|
| 3893/26400 [00:01<00:07, 3008.75it/s]
 16%|
| 4192/26400 [00:01<00:07, 2987.55it/s]
 17%|
| 4495/26400 [00:01<00:07, 2993.66it/s]
 18%|
| 4806/26400 [00:01<00:07, 3021.19it/s]
 19%|
| 5114/26400 [00:01<00:07, 3031.87it/s]
 21%|
| 5435/26400 [00:01<00:06, 3076.89it/s]
 22%|
| 5756/26400 [00:01<00:06, 3108.89it/s]
 23%|
| 6072/26400 [00:02<00:06, 3117.32it/s]
 24%|
| 6384/26400 [00:02<00:06, 3056.70it/s]
 25%|
| 6693/26400 [00:02<00:06, 3059.96it/s]
 27%|
| 7015/26400 [00:02<00:06, 3099.73it/s]
 28%|
| 7326/26400 [00:02<00:06, 2997.81it/s]
 29%|
| 7640/26400 [00:02<00:06, 3032.66it/s]
 30%|
| 7953/26400 [00:02<00:06, 3054.68it/s]
 31%|
| 8260/26400 [00:02<00:05, 3052.33it/s]
 32%|
```

```
| 8571/26400 [00:02<00:05, 3063.04it/s]
34%|█████████████████████████
| 8888/26400 [00:02<00:05, 3087.75it/s]
35%|██████████████████████████
| 9202/26400 [00:03<00:05, 3096.53it/s]
36%|███████████████████████████
| 9512/26400 [00:03<00:05, 3027.48it/s]
37%|████████████████████████████
| 9829/26400 [00:03<00:05, 3062.33it/s]
38%|████████████████████████████
| 10144/26400 [00:03<00:05, 3081.65it/s]
40%|█████████████████████████████
| 10461/26400 [00:03<00:05, 3100.97it/s]
41%|██████████████████████████████
| 10772/26400 [00:03<00:05, 3096.78it/s]
42%|███████████████████████████████
| 11087/26400 [00:03<00:04, 3105.95it/s]
43%|████████████████████████████████
| 11399/26400 [00:03<00:04, 3103.37it/s]
44%|█████████████████████████████████
| 11711/26400 [00:03<00:04, 3101.57it/s]
46%|██████████████████████████████████
| 12031/26400 [00:03<00:04, 3123.76it/s]
47%|██████████████████████████████████
| 12344/26400 [00:04<00:04, 3082.05it/s]
48%|███████████████████████████████████
| 12653/26400 [00:04<00:04, 3077.51it/s]
49%|████████████████████████████████████
| 12961/26400 [00:04<00:04, 3044.46it/s]
50%|█████████████████████████████████████
| 13273/26400 [00:04<00:04, 3060.15it/s]
51%|██████████████████████████████████████
| 13583/26400 [00:04<00:04, 3065.32it/s]
53%|███████████████████████████████████████
| 13907/26400 [00:04<00:04, 3109.16it/s]
54%|████████████████████████████████████████
| 14221/26400 [00:04<00:03, 3111.59it/s]
55%|█████████████████████████████████████████
| 14539/26400 [00:04<00:03, 3125.05it/s]
56%|██████████████████████████████████████████
| 14859/26400 [00:04<00:03, 3140.40it/s]
57%|██████████████████████████████████████████
| 15176/26400 [00:04<00:03, 3142.38it/s]
59%|███████████████████████████████████████████
| 15491/26400 [00:05<00:03, 3100.83it/s]
60%|████████████████████████████████████████████
| 15802/26400 [00:05<00:03, 3096.80it/s]
61%|█████████████████████████████████████████████
| 16112/26400 [00:05<00:03, 3072.65it/s]
62%|██████████████████████████████████████████████
| 16420/26400 [00:05<00:03, 3067.95it/s]
63%|███████████████████████████████████████████████
| 16727/26400 [00:05<00:03, 3052.93it/s]
65%|████████████████████████████████████████████████
| 17033/26400 [00:05<00:03, 3012.47it/s]
66%|████████████████████████████████████████████████
| 17338/26400 [00:05<00:03, 3017.07it/s]
```

```
 67%|
| 17655/26400 [00:05<00:02, 3054.91it/s]
 68%|
| 17967/26400 [00:05<00:02, 3067.53it/s]
 69%|
| 18282/26400 [00:05<00:02, 3084.88it/s]
 70%|
| 18604/26400 [00:06<00:02, 3117.88it/s]
 72%|
| 18922/26400 [00:06<00:02, 3129.49it/s]
 73%|
| 19236/26400 [00:06<00:02, 3107.29it/s]
 74%|
| 19549/26400 [00:06<00:02, 3107.29it/s]
 75%|
| 19860/26400 [00:06<00:02, 3101.32it/s]
 76%|
| 20171/26400 [00:06<00:02, 3087.94it/s]
 78%|
| 20486/26400 [00:06<00:01, 3099.60it/s]
 79%|
| 20804/26400 [00:06<00:01, 3116.58it/s]
 80%|
| 21130/26400 [00:06<00:01, 3151.57it/s]
 81%|
| 21450/26400 [00:06<00:01, 3159.09it/s]
 82%|
| 21767/26400 [00:07<00:01, 3155.45it/s]
 84%|
| 22083/26400 [00:07<00:01, 3140.55it/s]
 85%|
| 22398/26400 [00:07<00:01, 3136.51it/s]
 86%|
| 22712/26400 [00:07<00:01, 3093.80it/s]
 87%|
| 23022/26400 [00:07<00:01, 3079.71it/s]
 88%|
| 23331/26400 [00:07<00:00, 3076.07it/s]
 90%|
| 23652/26400 [00:07<00:00, 3108.45it/s]
 91%|
| 23978/26400 [00:07<00:00, 3145.74it/s]
 92%|
| 24296/26400 [00:07<00:00, 3149.09it/s]
 93%|
| 24612/26400 [00:08<00:00, 3136.14it/s]
 94%|
| 24926/26400 [00:08<00:00, 3130.44it/s]
 96%|
| 25249/26400 [00:08<00:00, 3152.80it/s]
 97%|
| 25565/26400 [00:08<00:00, 3148.17it/s]
 98%|
▌ | 25880/26400 [00:08<00:00, 3113.95it/s]
100%|
██| 26400/26400 [00:08<00:00, 3074.05it/s]
```

In [108]:
```
# avg w2v for project_titles
avg_w2v_vectors_pj_title_train = []; # the avg-w2v for each sentence/review is
stored in this list
for sentence in tqdm(X_train['project_title'].values): # for each review/sente
nce
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_pj_title_train.append(vector)

print(len(avg_w2v_vectors_pj_title_train))
print(len(avg_w2v_vectors_pj_title_train[0]))
print(avg_w2v_vectors_pj_title_train[0])
```

```
  0%|
| 0/35912 [00:00<?, ?it/s]
 41%|
| 14736/35912 [00:00<00:00, 146291.46it/s]
100%|
| 35912/35912 [00:00<00:00, 148181.44it/s]
```

```
35912
300
[-6.3004e-02  3.1239e-02 -1.8579e-01  4.4635e-02 -1.3728e-01  2.0852e-01
 -3.0928e+00  5.1131e-01  3.0234e-02 -5.0480e-01  1.8275e-01 -1.7641e-01
 -5.2801e-02  1.8643e-01 -1.5230e-01 -5.1885e-02  1.0524e-01  1.1579e-02
  2.4091e-01  1.0702e-02  6.3483e-03 -5.2369e-03 -9.6554e-02 -6.0413e-02
 -9.9636e-02 -4.6387e-02 -4.1514e-01  1.3550e-01  6.4773e-02 -1.5405e-01
 -9.3040e-01  4.1329e-01 -6.9841e-02 -1.2174e-02  2.4677e-01 -3.4287e-01
  3.9454e-01 -2.0151e-01  3.9742e-01 -5.3066e-02 -4.7727e-01 -4.1655e-01
  3.5602e-01  2.1111e-01  1.0663e-01 -4.1312e-01  8.2616e-03 -3.8526e-01
 -1.6343e-01  6.6813e-01 -1.9065e-01  3.3698e-01 -4.3189e-01  5.8042e-02
  5.5317e-01  3.4748e-01  6.4496e-02  2.2694e-03  2.0072e-01  2.2098e-01
 -3.0341e-01 -2.8312e-01 -3.1363e-01  5.3067e-01 -2.5296e-02 -1.5465e-01
  3.6343e-02 -4.6435e-01  2.3240e-01  1.8428e-01 -1.2815e-01 -1.3027e-02
 -1.9626e-01 -3.1234e-01 -7.1010e-02 -7.8771e-01 -2.9466e-01 -1.7576e-01
  6.7903e-02 -3.9181e-01  2.6900e-01 -6.8174e-01  2.5190e-01 -2.4059e-01
  2.7325e-01 -1.4917e-01  8.3279e-01 -3.0103e-01 -2.8582e-02 -3.3734e-01
 -2.9910e-01 -2.0228e-01  1.4101e-01  2.3135e-03 -2.7270e-01 -5.7234e-01
 -3.2172e+00 -4.6713e-01 -2.0993e-01 -4.8806e-01  4.9618e-01  7.8517e-02
  6.6969e-01  1.0868e-01  1.9829e-01  1.5625e-01  2.7715e-01  7.0318e-01
 -2.9784e-01  2.5801e-01 -3.4829e-01 -4.3414e-01  9.9575e-02  4.7150e-01
 -3.1241e-01  4.5226e-01  7.6259e-01  3.6508e-01  2.5221e-01 -1.2493e-01
  9.8863e-02  6.2592e-03 -2.9063e-01  3.1175e-01  3.8153e-01  4.8195e-01
  1.0216e-01 -1.0257e-01 -2.5863e-02 -3.7701e-01  8.7649e-02 -1.3177e-01
 -1.1675e-01  4.5153e-01  1.4577e-01  1.9500e-01  1.2384e-01  9.8835e-02
  5.3697e-01  4.9282e-01  6.9783e-02  7.0670e-02 -1.4149e-01  5.8837e-02
  6.3967e-02 -1.9756e-01 -3.6390e-01  1.0592e-01  1.6831e-01  6.1402e-01
 -8.0258e-02  3.4805e-01 -1.1233e-01 -1.4224e-01  8.6569e-03  1.5688e-01
  3.5244e-01  2.2691e-01 -5.0237e-02 -2.4736e-01 -2.8841e-01 -8.3235e-02
 -1.5517e-01 -3.5509e-01 -1.4373e-01  2.6160e-01  1.8030e-01 -3.8211e-02
 -3.8005e-01 -8.3100e-01 -3.3748e-01  3.7304e-02 -1.9378e-01 -5.4925e-01
 -7.3450e-02  1.3146e-01 -1.0388e+00  3.5216e-01  5.0045e-01 -1.7853e-01
  9.6374e-02 -9.7410e-02  6.3969e-01 -8.9913e-02  1.7378e-01 -9.7138e-02
 -2.9316e-01 -1.9741e-02 -2.8150e-01  7.4473e-01 -2.2485e-01  1.9390e-01
  2.9590e-02 -4.0950e-02  7.4386e-02 -8.6166e-02  2.5382e-01  1.1339e-01
  1.1157e-01  7.2497e-01 -6.2336e-01 -7.8740e-02 -1.9173e-01  1.7641e-01
  6.7506e-02 -3.6622e-01 -5.8263e-02  1.7013e-01  5.0655e-01  1.4760e-01
  3.5870e-01 -2.2899e-01  1.1184e-01 -2.1720e-02 -2.2123e-01 -3.1221e-01
 -6.3867e-02 -2.2743e-01 -4.0240e-02  4.5925e-01 -4.4871e-02  4.3306e-02
 -3.1833e-01  1.7021e-01 -3.5768e+00  3.8211e-01  4.8462e-02 -1.0886e-01
 -1.6740e-01  1.1626e-01  3.4908e-02  2.2989e-01  2.4380e-01  3.4586e-01
  1.5139e-01  6.6616e-01 -4.0853e-01  3.7251e-01 -6.9017e-01 -2.1990e-01
  4.4070e-01 -1.8349e-01 -3.1073e-01  1.8793e-01 -3.3471e-01  3.3364e-02
  3.1918e-02 -9.5115e-01 -1.7393e-01 -1.2559e-01  1.0029e-02 -1.4839e-02
  9.8882e-02 -1.7765e-01  1.9029e-02 -3.3748e-01 -4.4808e-01 -1.8664e-01
  7.2643e-02  1.9137e-01  9.5844e-01 -3.0384e-03  3.5579e-02 -1.8228e-01
 -5.5415e-01 -3.2349e-01 -3.7264e-01  3.3125e-01  5.4911e-01  1.5412e-01
  2.1926e-01 -1.5290e-01 -1.9009e-01  3.4024e-02  3.3934e-01  2.8025e-01
  1.0679e-01  2.9297e-01 -5.1538e-02 -4.7361e-02 -2.1209e-01  2.5042e-01
 -9.6005e-02  1.8085e-01  1.4923e-01  4.3730e-01 -3.0703e-01  3.2686e-01
  9.3145e-02  2.9420e-01 -2.5635e-01 -3.2371e-01 -2.8559e-01 -5.0285e-01
 -6.1455e-02 -3.5492e-01 -2.7389e-01 -1.5301e-01  5.3436e-01 -1.3498e-01]
```

In [109]:
```
avg_w2v_vectors_pj_title_cv = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_cv['project_title'].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_pj_title_cv.append(vector)
```

```
  0%|
| 0/17688 [00:00<?, ?it/s]
100%|
| 17688/17688 [00:00<00:00, 146571.93it/s]
```

In [110]:
```
avg_w2v_vectors_pj_title_test = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_test['project_title'].values): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    cnt_words =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if word in glove_words:
            vector += model[word]
            cnt_words += 1
    if cnt_words != 0:
        vector /= cnt_words
    avg_w2v_vectors_pj_title_test.append(vector)
```

```
  0%|
| 0/26400 [00:00<?, ?it/s]
100%|
| 26400/26400 [00:00<00:00, 153010.79it/s]
```

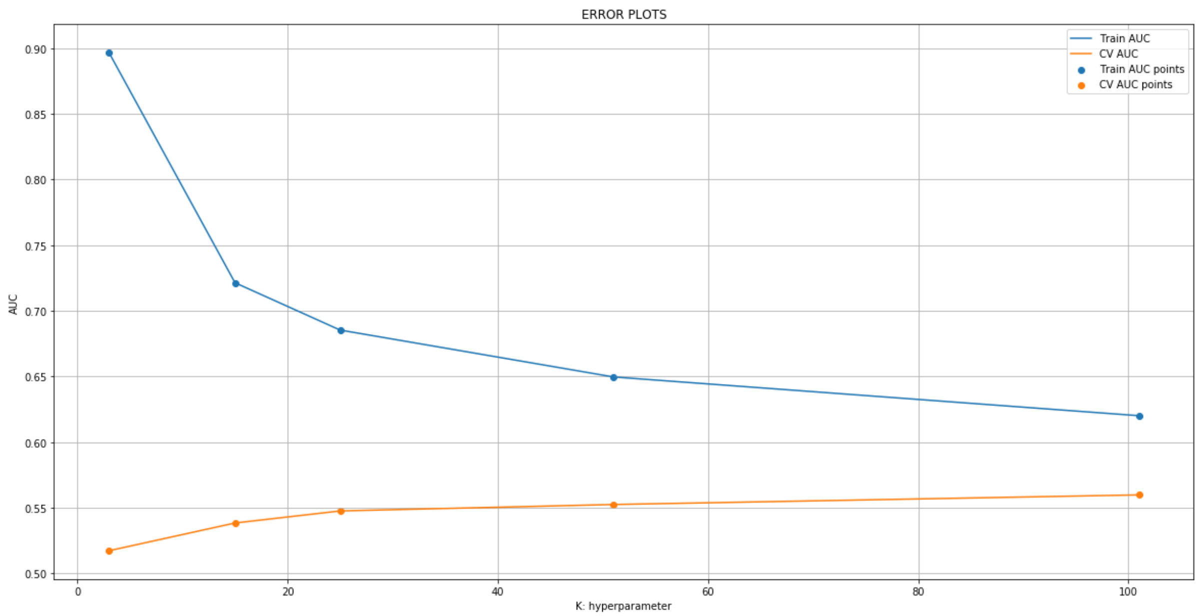
In [111]:
```
X_tr = hstack((avg_w2v_vectors_train, X_train_state_ohe, X_train_teacher_ohe,
               X_train_grade_ohe, X_train_category_ohe,
               X_train_subcategory_ohe, X_train_price_norm,
               X_train_teach_prev_norm, avg_w2v_vectors_pj_title_train)).tocsr
()

X_cr = hstack((avg_w2v_vectors_cv, X_cv_state_ohe, X_cv_teacher_ohe,
               X_cv_grade_ohe,X_cv_category_ohe,
               X_cv_subcategory_ohe, X_cv_price_norm,
              X_cv_teach_prev_norm, avg_w2v_vectors_pj_title_cv)).tocsr()

X_te = hstack((avg_w2v_vectors_test, X_test_state_ohe, X_test_teacher_ohe,
               X_test_grade_ohe, X_test_category_ohe,
               X_test_subcategory_ohe, X_test_price_norm,
              X_test_teach_prev_norm, avg_w2v_vectors_pj_title_test)).tocsr()

print("Final Data matrix")
print(X_tr.shape, y_train.shape)
print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix
(35912, 701) (35912,)
(17688, 701) (17688,)
(26400, 701) (26400,)
==============================================================================
======================
```

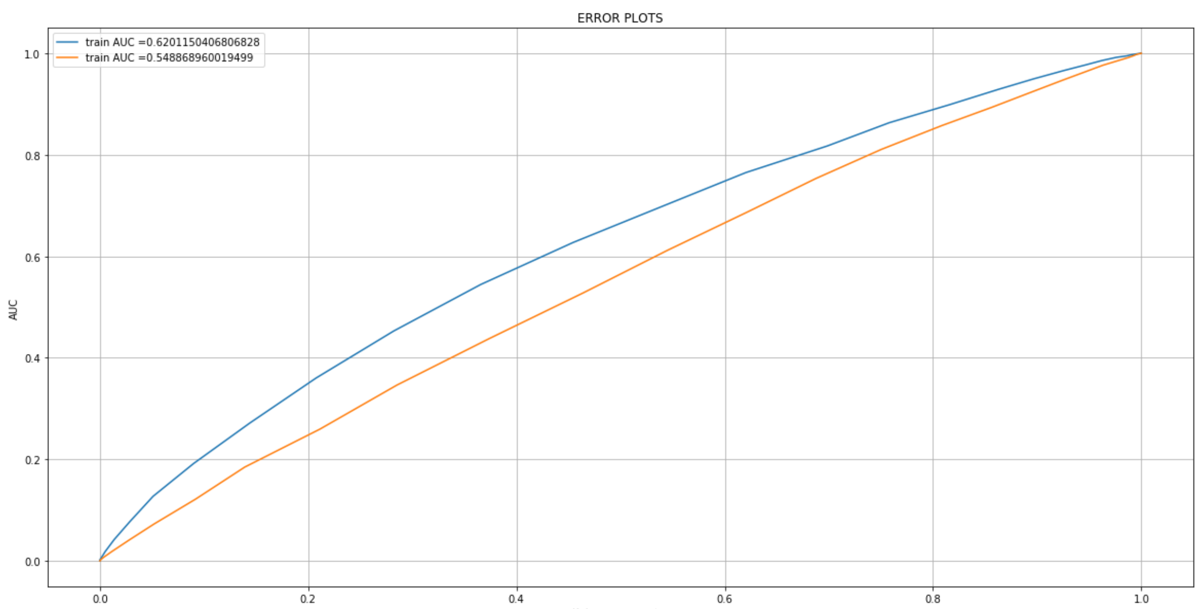
In [112]:
```python
train_auc = []
cv_auc = []
K = [3, 15, 25, 51, 101]
for i in tqdm(K):
    neigh = KNeighborsClassifier(n_neighbors=i, n_jobs=-1)
    neigh.fit(X_tr, y_train)

    y_train_pred = batch_predict(neigh, X_tr)
    y_cv_pred = batch_predict(neigh, X_cr)

    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
 20%|
| 1/5 [07:11<28:46, 431.50s/it]
 40%|
| 2/5 [14:31<21:42, 434.18s/it]
 60%|
| 3/5 [21:51<14:31, 435.65s/it]
 80%|
| 4/5 [29:10<07:16, 436.72s/it]
100%|
| 5/5 [36:30<00:00, 438.08s/it]
```

In [113]:
```
plt.figure(figsize=(20,10))
plt.plot(K, train_auc, label='Train AUC')
plt.plot(K, cv_auc, label='CV AUC')

plt.scatter(K, train_auc, label='Train AUC points')
plt.scatter(K, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



In [114]:
```
best_k = 101
from sklearn.metrics import roc_curve, auc


neigh = KNeighborsClassifier(n_neighbors=best_k, n_jobs=-1)
neigh.fit(X_tr, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estim
ates of the positive class
# not the predicted outputs

y_train_pred = batch_predict(neigh, X_tr)
y_test_pred = batch_predict(neigh, X_te)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
plt.figure(figsize=(20,10))
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tp
r)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



The Blue plot in the Graph is Train AUC and Orange Plot is Test AUC. Initially the label was accidentally copy pasted as Train AUC for both. Since redrawing plot would require to re run entire ipynb and it takes up a lot of time. so documenting the information as a markdown.

2.4.4 Applying KNN brute force on TFIDF W2V, SET 4

In [115]:
```
# Please write all the code with proper documentation
# preprocessing project_title and essay with TFIDF W2V Vectorization
tfidf_model = TfidfVectorizer()
tfidf_model.fit(X_train['essay'])
# we are converting a dictionary with word as a key, and the idf as a value
dictionary = dict(zip(tfidf_model.get_feature_names(), list(tfidf_model.idf_
)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [116]:
```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_train = []; # the avg-w2v for each sentence/review is stored
in this list
for sentence in tqdm(X_train['essay']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/revie
w
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf v
alue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_train.append(vector)

print(len(tfidf_w2v_vectors_train))
print(len(tfidf_w2v_vectors_train[0]))
```

```
  0%|
| 0/35912 [00:00<?, ?it/s]
  0%|
| 28/35912 [00:00<02:09, 277.97it/s]
  0%|
| 56/35912 [00:00<02:09, 276.33it/s]
  0%|▏
| 85/35912 [00:00<02:08, 278.89it/s]
  0%|▏
| 116/35912 [00:00<02:05, 284.60it/s]
  0%|▎
| 145/35912 [00:00<02:06, 283.08it/s]
  0%|▎
| 172/35912 [00:00<02:08, 277.54it/s]
  1%|▍
| 200/35912 [00:00<02:08, 277.67it/s]
  1%|▍
| 228/35912 [00:00<02:08, 276.92it/s]
  1%|▎
| 260/35912 [00:00<02:04, 285.71it/s]
  1%|▎
| 291/35912 [00:01<02:02, 291.17it/s]
  1%|▌
| 320/35912 [00:01<02:03, 288.45it/s]
  1%|▌
| 349/35912 [00:01<02:03, 288.28it/s]
  1%|▋
| 378/35912 [00:01<02:07, 278.24it/s]
  1%|▋
| 407/35912 [00:01<02:06, 280.25it/s]
  1%|▊
| 436/35912 [00:01<02:05, 281.69it/s]
  1%|▊
| 465/35912 [00:01<02:10, 272.37it/s]
  1%|▍
| 495/35912 [00:01<02:07, 277.22it/s]
  1%|▍
| 523/35912 [00:01<02:19, 253.39it/s]
  2%|▌
| 553/35912 [00:01<02:13, 264.57it/s]
  2%|▋
| 588/35912 [00:02<02:03, 284.97it/s]
  2%|▋
| 618/35912 [00:02<02:16, 258.22it/s]
  2%|▊
| 646/35912 [00:02<02:14, 263.10it/s]
  2%|▊
| 675/35912 [00:02<02:11, 268.58it/s]
  2%|▊
| 703/35912 [00:02<02:10, 270.55it/s]
  2%|▊
| 734/35912 [00:02<02:06, 278.46it/s]
  2%|█
| 767/35912 [00:02<02:00, 290.82it/s]
  2%|█
| 799/35912 [00:02<01:59, 294.28it/s]
  2%|█
```

```
| 830/35912 [00:02<01:59, 294.80it/s]
  2%|██
| 861/35912 [00:03<01:58, 296.86it/s]
  2%|██
| 891/35912 [00:03<02:00, 291.11it/s]
  3%|██
| 922/35912 [00:03<02:00, 289.30it/s]
  3%|██
| 952/35912 [00:03<01:59, 291.81it/s]
  3%|██
| 982/35912 [00:03<01:58, 293.59it/s]
  3%|██
| 1012/35912 [00:03<02:01, 286.42it/s]
  3%|██
| 1043/35912 [00:03<01:59, 291.68it/s]
  3%|██
| 1075/35912 [00:03<01:56, 298.17it/s]
  3%|██
| 1107/35912 [00:03<01:55, 302.07it/s]
  3%|██
| 1140/35912 [00:03<01:52, 308.44it/s]
  3%|██
| 1171/35912 [00:04<01:54, 304.61it/s]
  3%|██
| 1202/35912 [00:04<01:58, 293.43it/s]
  3%|██
| 1232/35912 [00:04<02:05, 275.31it/s]
  4%|███
| 1260/35912 [00:04<02:07, 272.09it/s]
  4%|███
| 1293/35912 [00:04<02:01, 285.18it/s]
  4%|███
| 1322/35912 [00:04<02:04, 277.00it/s]
  4%|███
| 1352/35912 [00:04<02:02, 282.14it/s]
  4%|███
| 1383/35912 [00:04<01:59, 288.56it/s]
  4%|███
| 1413/35912 [00:04<02:03, 278.33it/s]
  4%|███
| 1442/35912 [00:05<02:04, 276.33it/s]
  4%|███
| 1472/35912 [00:05<02:01, 282.43it/s]
  4%|███
| 1505/35912 [00:05<01:57, 292.29it/s]
  4%|███
| 1540/35912 [00:05<01:52, 306.11it/s]
  4%|███
| 1571/35912 [00:05<01:53, 303.88it/s]
  4%|███
| 1603/35912 [00:05<01:51, 307.02it/s]
  5%|███
| 1634/35912 [00:05<01:54, 300.13it/s]
  5%|███
| 1665/35912 [00:05<01:54, 298.03it/s]
  5%|███
| 1695/35912 [00:05<01:59, 286.06it/s]
```

```
  5%|████
| 1726/35912 [00:06<01:56, 292.24it/s]
  5%|████
| 1756/35912 [00:06<02:10, 261.00it/s]
  5%|████
| 1787/35912 [00:06<02:04, 273.46it/s]
  5%|████
| 1816/35912 [00:06<02:09, 264.03it/s]
  5%|████
| 1847/35912 [00:06<02:03, 275.04it/s]
  5%|████
| 1875/35912 [00:06<02:04, 274.31it/s]
  5%|████
| 1907/35912 [00:06<01:58, 286.02it/s]
  5%|████
| 1936/35912 [00:06<01:58, 286.58it/s]
  5%|████
| 1972/35912 [00:06<01:51, 303.92it/s]
  6%|████
| 2003/35912 [00:07<01:56, 291.34it/s]
  6%|████
| 2035/35912 [00:07<01:53, 298.77it/s]
  6%|████
| 2066/35912 [00:07<01:53, 298.80it/s]
  6%|████
| 2097/35912 [00:07<01:54, 294.57it/s]
  6%|████
| 2130/35912 [00:07<01:51, 303.76it/s]
  6%|████
| 2161/35912 [00:07<01:59, 282.50it/s]
  6%|█████
| 2197/35912 [00:07<01:52, 300.70it/s]
  6%|█████
| 2228/35912 [00:07<01:51, 302.78it/s]
  6%|█████
| 2260/35912 [00:07<01:49, 306.22it/s]
  6%|█████
| 2291/35912 [00:07<01:49, 305.77it/s]
  6%|█████
| 2322/35912 [00:08<01:53, 295.02it/s]
  7%|█████
| 2357/35912 [00:08<01:48, 308.20it/s]
  7%|█████
| 2389/35912 [00:08<01:58, 282.26it/s]
  7%|█████
| 2419/35912 [00:08<01:56, 286.75it/s]
  7%|█████
| 2454/35912 [00:08<01:50, 302.62it/s]
  7%|█████
| 2485/35912 [00:08<01:51, 300.61it/s]
  7%|█████
| 2516/35912 [00:08<01:55, 290.01it/s]
  7%|█████
| 2546/35912 [00:08<01:55, 288.11it/s]
  7%|█████
| 2582/35912 [00:08<01:50, 302.80it/s]
  7%|█████
```

```
| 2613/35912 [00:09<01:52, 297.30it/s]
  7%|█████▌
| 2644/35912 [00:09<01:54, 289.45it/s]
  7%|██████
| 2674/35912 [00:09<01:58, 280.48it/s]
  8%|██████
| 2706/35912 [00:09<01:54, 289.12it/s]
  8%|██████▎
| 2736/35912 [00:09<01:59, 277.93it/s]
  8%|██████▎
| 2767/35912 [00:09<01:56, 285.47it/s]
  8%|██████▌
| 2796/35912 [00:09<01:58, 279.58it/s]
  8%|██████▌
| 2825/35912 [00:09<01:59, 277.98it/s]
  8%|██████
| 2857/35912 [00:09<01:54, 288.81it/s]
  8%|██████
| 2887/35912 [00:10<01:59, 276.20it/s]
  8%|██████▏
| 2916/35912 [00:10<01:58, 279.61it/s]
  8%|██████▏
| 2945/35912 [00:10<02:04, 265.80it/s]
  8%|██████▍
| 2972/35912 [00:10<02:04, 264.91it/s]
  8%|██████▍
| 3002/35912 [00:10<02:00, 273.99it/s]
  8%|██████▋
| 3030/35912 [00:10<02:01, 269.62it/s]
  9%|██████▋
| 3058/35912 [00:10<02:05, 261.43it/s]
  9%|██████▍
| 3092/35912 [00:10<01:57, 279.02it/s]
  9%|██████▍
| 3125/35912 [00:10<01:52, 292.01it/s]
  9%|███████▏
| 3158/35912 [00:10<01:49, 298.58it/s]
  9%|███████▏
| 3189/35912 [00:11<01:53, 287.08it/s]
  9%|███████▎
| 3219/35912 [00:11<01:52, 289.39it/s]
  9%|███████▌
| 3249/35912 [00:11<01:56, 279.66it/s]
  9%|███████▌
| 3278/35912 [00:11<02:00, 271.03it/s]
  9%|███████▌
| 3306/35912 [00:11<02:02, 266.85it/s]
  9%|██████▉
| 3333/35912 [00:11<02:06, 256.57it/s]
  9%|██████▉
| 3359/35912 [00:11<02:08, 252.55it/s]
  9%|███████▏
| 3390/35912 [00:11<02:02, 265.54it/s]
 10%|███████▏
| 3417/35912 [00:11<02:05, 258.66it/s]
 10%|███████▍
| 3445/35912 [00:12<02:03, 263.42it/s]
```

```
 10%|███▏
| 3472/35912 [00:12<02:10, 249.42it/s]
 10%|███▏
| 3499/35912 [00:12<02:07, 254.01it/s]
 10%|███▏
| 3533/35912 [00:12<02:00, 269.82it/s]
 10%|███▍
| 3563/35912 [00:12<01:57, 276.10it/s]
 10%|███▍
| 3593/35912 [00:12<01:54, 281.51it/s]
 10%|███▌
| 3625/35912 [00:12<01:51, 289.10it/s]
 10%|███▌
| 3655/35912 [00:12<01:51, 289.14it/s]
 10%|███▋
| 3685/35912 [00:12<01:53, 283.45it/s]
 10%|███▋
| 3714/35912 [00:13<01:58, 272.75it/s]
 10%|███▊
| 3751/35912 [00:13<01:49, 294.87it/s]
 11%|███▌
| 3782/35912 [00:13<01:52, 284.67it/s]
 11%|███▌
| 3811/35912 [00:13<01:52, 284.79it/s]
 11%|███▋
| 3840/35912 [00:13<01:54, 279.94it/s]
 11%|███▋
| 3869/35912 [00:13<01:55, 277.43it/s]
 11%|███▋
| 3897/35912 [00:13<02:03, 258.42it/s]
 11%|███▊
| 3924/35912 [00:13<02:04, 257.52it/s]
 11%|███▊
| 3954/35912 [00:13<01:59, 268.41it/s]
 11%|███▉
| 3983/35912 [00:14<01:57, 272.44it/s]
 11%|███▉
| 4015/35912 [00:14<01:52, 283.09it/s]
 11%|███▊
| 4044/35912 [00:14<01:52, 283.68it/s]
 11%|███▊
| 4073/35912 [00:14<01:53, 281.62it/s]
 11%|████
| 4105/35912 [00:14<01:50, 288.40it/s]
 12%|████
| 4134/35912 [00:14<01:55, 275.94it/s]
 12%|████▏
| 4168/35912 [00:14<01:49, 290.43it/s]
 12%|████▏
| 4199/35912 [00:14<01:47, 295.42it/s]
 12%|████▏
| 4229/35912 [00:14<01:48, 291.83it/s]
 12%|████
| 4259/35912 [00:14<01:51, 284.44it/s]
 12%|████
| 4293/35912 [00:15<01:46, 296.19it/s]
 12%|████▏
```

```
| 4323/35912 [00:15<01:47, 293.21it/s]
 12%|█████████
| 4353/35912 [00:15<01:50, 285.36it/s]
 12%|█████████▍
| 4388/35912 [00:15<01:44, 301.53it/s]
 12%|█████████▍
| 4419/35912 [00:15<01:46, 296.42it/s]
 12%|█████████▋
| 4453/35912 [00:15<01:42, 307.66it/s]
 12%|█████████▋
| 4485/35912 [00:15<01:47, 292.75it/s]
 13%|█████████▌
| 4517/35912 [00:15<01:45, 296.48it/s]
 13%|██████████
| 4550/35912 [00:15<01:43, 303.50it/s]
 13%|██████████
| 4581/35912 [00:16<01:46, 295.21it/s]
 13%|██████████▎
| 4615/35912 [00:16<01:42, 305.92it/s]
 13%|██████████▎
| 4646/35912 [00:16<01:45, 296.81it/s]
 13%|██████████▌
| 4676/35912 [00:16<01:46, 294.47it/s]
 13%|██████████▌
| 4706/35912 [00:16<01:46, 292.05it/s]
 13%|██████████
| 4736/35912 [00:16<01:53, 273.70it/s]
 13%|██████████
| 4770/35912 [00:16<01:47, 288.70it/s]
 13%|██████████▏
| 4800/35912 [00:16<01:47, 289.69it/s]
 13%|██████████▏
| 4830/35912 [00:16<01:50, 281.43it/s]
 14%|██████████▍
| 4860/35912 [00:17<01:48, 285.34it/s]
 14%|██████████▍
| 4889/35912 [00:17<01:50, 279.50it/s]
 14%|██████████▋
| 4918/35912 [00:17<01:51, 277.93it/s]
 14%|██████████▋
| 4951/35912 [00:17<01:46, 290.41it/s]
 14%|██████████▍
| 4985/35912 [00:17<01:42, 301.49it/s]
 14%|██████████▍
| 5019/35912 [00:17<01:39, 308.93it/s]
 14%|███████████
| 5051/35912 [00:17<01:45, 293.55it/s]
 14%|███████████▎
| 5083/35912 [00:17<01:42, 300.39it/s]
 14%|███████████▎
| 5114/35912 [00:17<01:44, 294.82it/s]
 14%|███████████▌
| 5145/35912 [00:17<01:43, 296.02it/s]
 14%|███████████▌
| 5181/35912 [00:18<01:38, 310.49it/s]
 15%|███████████
| 5213/35912 [00:18<01:38, 311.70it/s]
```

```
 15%|███████████
| 5246/35912 [00:18<01:37, 314.50it/s]
 15%|███████████▏
| 5278/35912 [00:18<01:37, 313.60it/s]
 15%|███████████▏
| 5310/35912 [00:18<01:39, 306.69it/s]
 15%|███████████▎
| 5341/35912 [00:18<01:45, 289.01it/s]
 15%|███████████▎
| 5371/35912 [00:18<01:45, 289.09it/s]
 15%|███████████▍
| 5401/35912 [00:18<01:51, 273.37it/s]
 15%|███████████▍
| 5432/35912 [00:18<01:48, 282.08it/s]
 15%|███████████▌
| 5461/35912 [00:19<01:51, 274.16it/s]
 15%|███████████▌
| 5489/35912 [00:19<01:50, 275.30it/s]
 15%|███████████▋
| 5524/35912 [00:19<01:43, 292.85it/s]
 15%|███████████▊
| 5556/35912 [00:19<01:42, 296.56it/s]
 16%|███████████▊
| 5589/35912 [00:19<01:39, 305.24it/s]
 16%|███████████▉
| 5620/35912 [00:19<01:42, 296.36it/s]
 16%|███████████▉
| 5650/35912 [00:19<01:44, 288.26it/s]
 16%|████████████
| 5683/35912 [00:19<01:41, 299.04it/s]
 16%|████████████
| 5714/35912 [00:19<01:40, 300.72it/s]
 16%|████████████▏
| 5745/35912 [00:20<01:41, 296.72it/s]
 16%|████████████▏
| 5778/35912 [00:20<01:38, 304.52it/s]
 16%|████████████▎
| 5809/35912 [00:20<01:40, 299.31it/s]
 16%|████████████▎
| 5840/35912 [00:20<01:40, 300.04it/s]
 16%|████████████▍
| 5871/35912 [00:20<01:39, 302.31it/s]
 16%|████████████▍
| 5903/35912 [00:20<01:37, 306.77it/s]
 17%|████████████▌
| 5934/35912 [00:20<01:38, 304.35it/s]
 17%|████████████▌
| 5965/35912 [00:20<01:40, 298.31it/s]
 17%|████████████▋
| 5996/35912 [00:20<01:39, 301.10it/s]
 17%|████████████▊
| 6027/35912 [00:20<01:40, 296.99it/s]
 17%|████████████▊
| 6057/35912 [00:21<01:48, 275.97it/s]
 17%|████████████▉
| 6085/35912 [00:21<01:50, 269.40it/s]
 17%|████████████▉
```

```
| 6116/35912 [00:21<01:48, 275.40it/s]
 17%|█████████████
| 6148/35912 [00:21<01:44, 285.32it/s]
 17%|█████████████
| 6178/35912 [00:21<01:43, 286.48it/s]
 17%|█████████████▎
| 6207/35912 [00:21<01:44, 284.38it/s]
 17%|█████████████▎
| 6237/35912 [00:21<01:43, 287.46it/s]
 17%|█████████████▍
| 6268/35912 [00:21<01:41, 292.43it/s]
 18%|█████████████▍
| 6298/35912 [00:21<01:40, 294.03it/s]
 18%|█████████████▋
| 6328/35912 [00:22<01:42, 290.01it/s]
 18%|█████████████▋
| 6358/35912 [00:22<01:42, 288.97it/s]
 18%|█████████████▌
| 6387/35912 [00:22<01:43, 285.25it/s]
 18%|█████████████▌
| 6416/35912 [00:22<01:43, 285.19it/s]
 18%|██████████████
| 6448/35912 [00:22<01:40, 292.61it/s]
 18%|██████████████
| 6478/35912 [00:22<01:43, 284.96it/s]
 18%|██████████████▎
| 6514/35912 [00:22<01:36, 303.41it/s]
 18%|██████████████▎
| 6548/35912 [00:22<01:34, 309.48it/s]
 18%|██████████████▌
| 6580/35912 [00:22<01:38, 298.83it/s]
 18%|██████████████▌
| 6613/35912 [00:22<01:35, 306.07it/s]
 19%|██████████████
| 6644/35912 [00:23<01:36, 302.98it/s]
 19%|██████████████▏
| 6677/35912 [00:23<01:35, 304.83it/s]
 19%|██████████████▏
| 6713/35912 [00:23<01:32, 316.39it/s]
 19%|██████████████▍
| 6745/35912 [00:23<01:36, 302.44it/s]
 19%|██████████████▍
| 6776/35912 [00:23<01:38, 297.04it/s]
 19%|██████████████▋
| 6809/35912 [00:23<01:35, 304.75it/s]
 19%|██████████████▋
| 6844/35912 [00:23<01:32, 315.57it/s]
 19%|██████████████▍
| 6876/35912 [00:23<01:31, 316.19it/s]
 19%|██████████████▍
| 6908/35912 [00:23<01:40, 289.24it/s]
 19%|███████████████
| 6944/35912 [00:24<01:35, 302.17it/s]
 19%|███████████████▎
| 6975/35912 [00:24<01:38, 294.33it/s]
 20%|███████████████▎
| 7005/35912 [00:24<01:44, 275.86it/s]
```

```
 20%|███████▍
| 7034/35912 [00:24<01:47, 269.28it/s]
 20%|███████▍
| 7071/35912 [00:24<01:38, 291.34it/s]
 20%|███████
| 7101/35912 [00:24<01:38, 291.55it/s]
 20%|███████
| 7132/35912 [00:24<01:37, 296.23it/s]
 20%|███████▏
| 7163/35912 [00:24<01:37, 293.65it/s]
 20%|███████▏
| 7193/35912 [00:24<01:39, 287.28it/s]
 20%|███████▏
| 7226/35912 [00:25<01:36, 295.90it/s]
 20%|███████▏
| 7256/35912 [00:25<01:39, 287.96it/s]
 20%|███████▎
| 7288/35912 [00:25<01:38, 289.85it/s]
 20%|███████▎
| 7318/35912 [00:25<01:38, 289.67it/s]
 20%|███████▍
| 7351/35912 [00:25<01:35, 297.65it/s]
 21%|███████▍
| 7381/35912 [00:25<01:36, 295.09it/s]
 21%|███████▌
| 7412/35912 [00:25<01:35, 298.77it/s]
 21%|███████▌
| 7442/35912 [00:25<01:36, 294.98it/s]
 21%|███████▋
| 7472/35912 [00:25<01:38, 289.00it/s]
 21%|███████▋
| 7508/35912 [00:25<01:32, 306.60it/s]
 21%|███████▋
| 7540/35912 [00:26<01:35, 296.94it/s]
 21%|███████▋
| 7570/35912 [00:26<01:37, 289.48it/s]
 21%|███████▋
| 7600/35912 [00:26<01:38, 286.10it/s]
 21%|███████▋
| 7629/35912 [00:26<01:39, 284.12it/s]
 21%|███████▋
| 7658/35912 [00:26<01:39, 282.75it/s]
 21%|███████▊
| 7687/35912 [00:26<01:41, 279.34it/s]
 21%|███████▊
| 7716/35912 [00:26<01:44, 270.83it/s]
 22%|███████▊
| 7750/35912 [00:26<01:37, 287.91it/s]
 22%|███████▊
| 7782/35912 [00:26<01:34, 296.24it/s]
 22%|███████▉
| 7812/35912 [00:27<01:38, 284.10it/s]
 22%|███████▉
| 7841/35912 [00:27<01:41, 275.50it/s]
 22%|████████
| 7869/35912 [00:27<01:42, 273.01it/s]
 22%|████████
```

```
| 7900/35912 [00:27<01:39, 281.05it/s]
 22%|████████████████▋
| 7929/35912 [00:27<01:40, 278.19it/s]
 22%|████████████████▋
| 7957/35912 [00:27<01:45, 265.50it/s]
 22%|████████████████▊
| 7992/35912 [00:27<01:37, 285.02it/s]
 22%|████████████████▊
| 8022/35912 [00:27<01:38, 283.04it/s]
 22%|█████████████████
| 8053/35912 [00:27<01:36, 289.22it/s]
 23%|█████████████████
| 8089/35912 [00:27<01:30, 305.99it/s]
 23%|█████████████████▏
| 8121/35912 [00:28<01:30, 307.63it/s]
 23%|█████████████████▎
| 8153/35912 [00:28<01:33, 295.96it/s]
 23%|█████████████████▎
| 8185/35912 [00:28<01:31, 302.16it/s]
 23%|█████████████████▍
| 8217/35912 [00:28<01:31, 303.18it/s]
 23%|█████████████████▍
| 8248/35912 [00:28<01:34, 293.33it/s]
 23%|█████████████████▌
| 8278/35912 [00:28<01:34, 292.94it/s]
 23%|█████████████████▌
| 8308/35912 [00:28<01:35, 289.29it/s]
 23%|█████████████████▊
| 8338/35912 [00:28<01:35, 289.27it/s]
 23%|█████████████████▊
| 8368/35912 [00:28<01:40, 274.23it/s]
 23%|█████████████████▉
| 8400/35912 [00:29<01:36, 285.96it/s]
 23%|█████████████████▉
| 8435/35912 [00:29<01:31, 301.22it/s]
 24%|██████████████████
| 8469/35912 [00:29<01:29, 307.06it/s]
 24%|██████████████████
| 8501/35912 [00:29<01:36, 284.59it/s]
 24%|██████████████████
| 8533/35912 [00:29<01:33, 292.16it/s]
 24%|██████████████████
| 8563/35912 [00:29<01:34, 289.59it/s]
 24%|██████████████████▏
| 8593/35912 [00:29<01:34, 287.82it/s]
 24%|██████████████████▏
| 8623/35912 [00:29<01:34, 289.90it/s]
 24%|██████████████████▎
| 8658/35912 [00:29<01:29, 303.49it/s]
 24%|██████████████████▍
| 8689/35912 [00:30<01:35, 286.23it/s]
 24%|██████████████████▍
| 8719/35912 [00:30<01:34, 286.31it/s]
 24%|██████████████████▌
| 8749/35912 [00:30<01:34, 287.18it/s]
 24%|██████████████████▌
| 8778/35912 [00:30<01:39, 272.81it/s]
```

```
 25%|███████▋
| 8812/35912 [00:30<01:34, 288.02it/s]
 25%|███████▋
| 8845/35912 [00:30<01:30, 298.04it/s]
 25%|███████▋
| 8876/35912 [00:30<01:30, 300.02it/s]
 25%|███████▋
| 8907/35912 [00:30<01:32, 291.25it/s]
 25%|███████▊
| 8940/35912 [00:30<01:29, 301.28it/s]
 25%|███████▊
| 8971/35912 [00:31<01:34, 284.85it/s]
 25%|████████
| 9002/35912 [00:31<01:32, 289.73it/s]
 25%|████████
| 9035/35912 [00:31<01:29, 299.32it/s]
 25%|████████
| 9066/35912 [00:31<01:30, 296.62it/s]
 25%|████████
| 9096/35912 [00:31<01:36, 277.27it/s]
 25%|████████▏
| 9125/35912 [00:31<01:35, 280.36it/s]
 25%|████████▏
| 9154/35912 [00:31<01:35, 279.34it/s]
 26%|████████▏
| 9183/35912 [00:31<01:35, 280.22it/s]
 26%|████████▏
| 9213/35912 [00:31<01:34, 282.87it/s]
 26%|████████▎
| 9242/35912 [00:31<01:36, 275.47it/s]
 26%|████████▍
| 9274/35912 [00:32<01:33, 286.14it/s]
 26%|████████▍
| 9303/35912 [00:32<01:34, 282.48it/s]
 26%|████████▍
| 9332/35912 [00:32<01:35, 279.18it/s]
 26%|████████▌
| 9361/35912 [00:32<01:36, 274.55it/s]
 26%|████████▌
| 9389/35912 [00:32<01:36, 275.57it/s]
 26%|████████▋
| 9424/35912 [00:32<01:30, 293.81it/s]
 26%|████████▍
| 9454/35912 [00:32<01:30, 293.28it/s]
 26%|████████▍
| 9484/35912 [00:32<01:33, 281.40it/s]
 26%|████████▌
| 9515/35912 [00:32<01:31, 288.01it/s]
 27%|████████▌
| 9547/35912 [00:33<01:29, 294.68it/s]
 27%|████████▌
| 9577/35912 [00:33<01:29, 294.75it/s]
 27%|████████▌
| 9607/35912 [00:33<01:29, 293.07it/s]
 27%|████████▋
| 9637/35912 [00:33<01:34, 278.14it/s]
 27%|████████▋
```

```
| 9669/35912 [00:33<01:31, 287.38it/s]
 27%|████████████████████▌
| 9700/35912 [00:33<01:29, 291.55it/s]
 27%|████████████████████▌
| 9730/35912 [00:33<01:29, 291.70it/s]
 27%|████████████████████▏▋
| 9760/35912 [00:33<01:29, 291.80it/s]
 27%|████████████████████▏▋
| 9792/35912 [00:33<01:28, 294.99it/s]
 27%|████████████████████▏▊
| 9822/35912 [00:33<01:29, 290.69it/s]
 27%|████████████████████▏▊
| 9852/35912 [00:34<01:32, 281.29it/s]
 28%|████████████████████▏█
| 9881/35912 [00:34<01:31, 283.24it/s]
 28%|████████████████████▏█
| 9910/35912 [00:34<01:37, 266.61it/s]
 28%|█████████████████████
| 9940/35912 [00:34<01:34, 274.51it/s]
 28%|█████████████████████
| 9968/35912 [00:34<01:34, 273.93it/s]
 28%|█████████████████████▏
| 9996/35912 [00:34<01:35, 271.14it/s]
 28%|████████████████████▏█
| 10024/35912 [00:34<01:39, 260.97it/s]
 28%|████████████████████▏█
| 10054/35912 [00:34<01:35, 269.57it/s]
 28%|█████████████████████
| 10088/35912 [00:34<01:30, 286.90it/s]
 28%|█████████████████████▏
| 10118/35912 [00:35<01:32, 280.35it/s]
 28%|█████████████████████▏
| 10147/35912 [00:35<01:36, 267.75it/s]
 28%|█████████████████████▍
| 10178/35912 [00:35<01:33, 274.92it/s]
 28%|█████████████████████▍
| 10206/35912 [00:35<01:37, 264.15it/s]
 28%|█████████████████████▍
| 10233/35912 [00:35<01:37, 262.22it/s]
 29%|█████████████████████▋
| 10265/35912 [00:35<01:32, 276.71it/s]
 29%|█████████████████████▌
| 10298/35912 [00:35<01:28, 288.72it/s]
 29%|█████████████████████▌
| 10328/35912 [00:35<01:31, 279.22it/s]
 29%|█████████████████████▏▉
| 10357/35912 [00:35<01:35, 268.50it/s]
 29%|█████████████████████▏▉
| 10385/35912 [00:36<01:34, 271.27it/s]
 29%|█████████████████████▏█
| 10415/35912 [00:36<01:31, 277.95it/s]
 29%|█████████████████████▏█
| 10447/35912 [00:36<01:29, 284.94it/s]
 29%|█████████████████████▏█▋
| 10476/35912 [00:36<01:29, 283.31it/s]
 29%|█████████████████████▏█▋
| 10505/35912 [00:36<01:34, 268.14it/s]
```

```
 29%|█████████████████████▍
| 10534/35912 [00:36<01:32, 273.77it/s]
 29%|██████████████████████
| 10562/35912 [00:36<01:36, 263.41it/s]
 29%|██████████████████████
| 10593/35912 [00:36<01:31, 275.31it/s]
 30%|██████████████████████▏
| 10623/35912 [00:36<01:29, 281.70it/s]
 30%|██████████████████████▏
| 10652/35912 [00:37<01:30, 279.44it/s]
 30%|██████████████████████▎
| 10682/35912 [00:37<01:28, 283.91it/s]
 30%|██████████████████████▎
| 10711/35912 [00:37<01:29, 280.96it/s]
 30%|██████████████████████▍
| 10745/35912 [00:37<01:25, 295.84it/s]
 30%|██████████████████████▌
| 10777/35912 [00:37<01:24, 298.69it/s]
 30%|██████████████████████▌
| 10808/35912 [00:37<01:26, 291.19it/s]
 30%|██████████████████████▋
| 10839/35912 [00:37<01:25, 294.29it/s]
 30%|██████████████████████▋
| 10871/35912 [00:37<01:23, 300.07it/s]
 30%|██████████████████████▊
| 10902/35912 [00:37<01:28, 281.02it/s]
 30%|██████████████████████▊
| 10932/35912 [00:37<01:27, 284.25it/s]
 31%|██████████████████████▉
| 10961/35912 [00:38<01:36, 258.68it/s]
 31%|██████████████████████▉
| 10991/35912 [00:38<01:32, 269.30it/s]
 31%|███████████████████████
| 11021/35912 [00:38<01:29, 277.27it/s]
 31%|███████████████████████
| 11054/35912 [00:38<01:25, 289.90it/s]
 31%|███████████████████████▏
| 11086/35912 [00:38<01:23, 297.71it/s]
 31%|███████████████████████▏
| 11117/35912 [00:38<01:27, 281.84it/s]
 31%|███████████████████████▎
| 11151/35912 [00:38<01:24, 294.22it/s]
 31%|███████████████████████▎
| 11183/35912 [00:38<01:22, 299.20it/s]
 31%|███████████████████████▍
| 11214/35912 [00:38<01:27, 282.01it/s]
 31%|███████████████████████▍
| 11243/35912 [00:39<01:27, 282.89it/s]
 31%|███████████████████████▌
| 11273/35912 [00:39<01:26, 284.79it/s]
 31%|███████████████████████▌
| 11302/35912 [00:39<01:26, 283.21it/s]
 32%|███████████████████████▋
| 11333/35912 [00:39<01:24, 290.15it/s]
 32%|███████████████████████▋
| 11366/35912 [00:39<01:21, 299.64it/s]
 32%|███████████████████████▊
```

```
| 11397/35912 [00:39<01:28, 277.00it/s]
 32%|
| 11428/35912 [00:39<01:26, 282.44it/s]
 32%|
| 11457/35912 [00:39<01:26, 282.40it/s]
 32%|
| 11486/35912 [00:39<01:30, 270.55it/s]
 32%|
| 11518/35912 [00:40<01:27, 280.19it/s]
 32%|
| 11552/35912 [00:40<01:23, 292.20it/s]
 32%|
| 11584/35912 [00:40<01:21, 299.40it/s]
 32%|
| 11617/35912 [00:40<01:20, 302.29it/s]
 32%|
| 11648/35912 [00:40<01:25, 283.92it/s]
 33%|
| 11684/35912 [00:40<01:20, 301.06it/s]
 33%|
| 11716/35912 [00:40<01:19, 303.26it/s]
 33%|
| 11750/35912 [00:40<01:17, 311.93it/s]
 33%|
| 11782/35912 [00:40<01:23, 289.84it/s]
 33%|
| 11812/35912 [00:41<01:25, 280.73it/s]
 33%|
| 11842/35912 [00:41<01:24, 284.03it/s]
 33%|
| 11871/35912 [00:41<01:25, 282.69it/s]
 33%|
| 11905/35912 [00:41<01:20, 296.40it/s]
 33%|
| 11939/35912 [00:41<01:18, 305.17it/s]
 33%|
| 11971/35912 [00:41<01:17, 308.79it/s]
 33%|
| 12003/35912 [00:41<01:19, 300.07it/s]
 34%|
| 12034/35912 [00:41<01:24, 284.10it/s]
 34%|
| 12064/35912 [00:41<01:23, 287.26it/s]
 34%|
| 12094/35912 [00:41<01:22, 290.35it/s]
 34%|
| 12125/35912 [00:42<01:22, 289.58it/s]
 34%|
| 12155/35912 [00:42<01:24, 281.35it/s]
 34%|
| 12184/35912 [00:42<01:32, 257.68it/s]
 34%|
| 12213/35912 [00:42<01:29, 263.88it/s]
 34%|
| 12242/35912 [00:42<01:27, 269.15it/s]
 34%|
| 12270/35912 [00:42<01:27, 271.73it/s]
```

```
 34%|█████████████████████████▏
| 12303/35912 [00:42<01:22, 284.91it/s]
 34%|█████████████████████████▍
| 12333/35912 [00:42<01:21, 288.66it/s]
 34%|█████████████████████████▍
| 12365/35912 [00:42<01:19, 295.97it/s]
 35%|█████████████████████████▌
| 12395/35912 [00:43<01:21, 289.67it/s]
 35%|█████████████████████████▌
| 12425/35912 [00:43<01:21, 289.54it/s]
 35%|██████████████████████████
| 12455/35912 [00:43<01:21, 289.45it/s]
 35%|██████████████████████████
| 12485/35912 [00:43<01:23, 282.06it/s]
 35%|██████████████████████████▏
| 12514/35912 [00:43<01:26, 271.09it/s]
 35%|██████████████████████████▏
| 12542/35912 [00:43<01:25, 272.32it/s]
 35%|██████████████████████████▍
| 12570/35912 [00:43<01:27, 266.20it/s]
 35%|██████████████████████████▍
| 12602/35912 [00:43<01:23, 279.80it/s]
 35%|██████████████████████████▌
| 12634/35912 [00:43<01:20, 287.84it/s]
 35%|██████████████████████████▌
| 12664/35912 [00:44<01:21, 286.61it/s]
 35%|██████████████████████████▌
| 12694/35912 [00:44<01:20, 288.22it/s]
 35%|██████████████████████████▌
| 12725/35912 [00:44<01:18, 293.81it/s]
 36%|███████████████████████████
| 12755/35912 [00:44<01:19, 291.57it/s]
 36%|███████████████████████████
| 12786/35912 [00:44<01:18, 296.24it/s]
 36%|███████████████████████████▍
| 12818/35912 [00:44<01:16, 302.36it/s]
 36%|███████████████████████████▍
| 12849/35912 [00:44<01:20, 286.32it/s]
 36%|███████████████████████████▌
| 12878/35912 [00:44<01:24, 273.83it/s]
 36%|███████████████████████████▌
| 12913/35912 [00:44<01:19, 290.96it/s]
 36%|███████████████████████████
| 12949/35912 [00:44<01:14, 307.36it/s]
 36%|███████████████████████████
| 12982/35912 [00:45<01:13, 313.17it/s]
 36%|███████████████████████████▏
| 13014/35912 [00:45<01:17, 297.04it/s]
 36%|███████████████████████████▏
| 13045/35912 [00:45<01:20, 285.29it/s]
 36%|███████████████████████████▍
| 13077/35912 [00:45<01:17, 293.47it/s]
 36%|███████████████████████████▍
| 13107/35912 [00:45<01:18, 291.36it/s]
 37%|███████████████████████████▌
| 13137/35912 [00:45<01:21, 279.37it/s]
 37%|███████████████████████████▌
```

```
| 13166/35912 [00:45<01:21, 279.44it/s]
 37%|███████████████████████████
| 13196/35912 [00:45<01:20, 283.10it/s]
 37%|███████████████████████████
| 13225/35912 [00:45<01:19, 284.53it/s]
 37%|███████████████████████████▌
| 13254/35912 [00:46<01:19, 283.83it/s]
 37%|███████████████████████████▌
| 13284/35912 [00:46<01:19, 283.84it/s]
 37%|███████████████████████████▌
| 13315/35912 [00:46<01:17, 290.61it/s]
 37%|███████████████████████████▌
| 13345/35912 [00:46<01:21, 278.11it/s]
 37%|███████████████████████████▋
| 13379/35912 [00:46<01:16, 293.63it/s]
 37%|████████████████████████████
| 13409/35912 [00:46<01:16, 293.13it/s]
 37%|████████████████████████████
| 13439/35912 [00:46<01:17, 291.12it/s]
 38%|████████████████████████████▏
| 13469/35912 [00:46<01:17, 288.05it/s]
 38%|████████████████████████████▏
| 13498/35912 [00:46<01:19, 282.14it/s]
 38%|████████████████████████████▎
| 13530/35912 [00:46<01:16, 291.14it/s]
 38%|████████████████████████████▎
| 13560/35912 [00:47<01:19, 282.36it/s]
 38%|████████████████████████████▍
| 13589/35912 [00:47<01:21, 275.15it/s]
 38%|████████████████████████████▍
| 13617/35912 [00:47<01:22, 271.19it/s]
 38%|████████████████████████████▍
| 13649/35912 [00:47<01:18, 282.89it/s]
 38%|████████████████████████████▍
| 13679/35912 [00:47<01:17, 285.58it/s]
 38%|████████████████████████████▌
| 13708/35912 [00:47<01:18, 282.10it/s]
 38%|████████████████████████████▌
| 13737/35912 [00:47<01:19, 280.53it/s]
 38%|████████████████████████████▋
| 13771/35912 [00:47<01:15, 293.97it/s]
 38%|████████████████████████████▋
| 13803/35912 [00:47<01:13, 299.86it/s]
 39%|████████████████████████████▊
| 13837/35912 [00:48<01:11, 310.25it/s]
 39%|████████████████████████████▊
| 13870/35912 [00:48<01:09, 315.26it/s]
 39%|█████████████████████████████
| 13902/35912 [00:48<01:10, 311.38it/s]
 39%|█████████████████████████████
| 13934/35912 [00:48<01:11, 306.95it/s]
 39%|█████████████████████████████▏
| 13968/35912 [00:48<01:09, 314.65it/s]
 39%|█████████████████████████████▏
| 14000/35912 [00:48<01:13, 297.97it/s]
 39%|█████████████████████████████▎
| 14031/35912 [00:48<01:12, 300.84it/s]
```

```
 39%|████████████████████████████▏
| 14062/35912 [00:48<01:12, 301.12it/s]
 39%|████████████████████████████▍
| 14093/35912 [00:48<01:15, 287.92it/s]
 39%|████████████████████████████▍
| 14124/35912 [00:48<01:14, 291.12it/s]
 39%|████████████████████████████▌
| 14160/35912 [00:49<01:11, 305.92it/s]
 40%|████████████████████████████▊
| 14191/35912 [00:49<01:13, 294.28it/s]
 40%|████████████████████████████▊
| 14224/35912 [00:49<01:11, 302.71it/s]
 40%|████████████████████████████▉
| 14255/35912 [00:49<01:12, 298.94it/s]
 40%|████████████████████████████▉
| 14286/35912 [00:49<01:12, 298.92it/s]
 40%|█████████████████████████████
| 14317/35912 [00:49<01:13, 292.16it/s]
 40%|█████████████████████████████
| 14355/35912 [00:49<01:08, 312.59it/s]
 40%|█████████████████████████████
| 14387/35912 [00:49<01:09, 308.66it/s]
 40%|█████████████████████████████
| 14419/35912 [00:49<01:15, 284.79it/s]
 40%|█████████████████████████████▏
| 14449/35912 [00:50<01:17, 277.40it/s]
 40%|█████████████████████████████▏
| 14478/35912 [00:50<01:16, 278.85it/s]
 40%|█████████████████████████████▎
| 14510/35912 [00:50<01:14, 287.91it/s]
 40%|█████████████████████████████▎
| 14540/35912 [00:50<01:16, 277.92it/s]
 41%|█████████████████████████████▍
| 14569/35912 [00:50<01:15, 280.84it/s]
 41%|█████████████████████████████▍
| 14599/35912 [00:50<01:15, 284.11it/s]
 41%|█████████████████████████████▌
| 14628/35912 [00:50<01:16, 277.86it/s]
 41%|█████████████████████████████▌
| 14659/35912 [00:50<01:14, 284.64it/s]
 41%|█████████████████████████████▋
| 14688/35912 [00:50<01:20, 264.56it/s]
 41%|█████████████████████████████▋
| 14715/35912 [00:51<01:20, 261.74it/s]
 41%|█████████████████████████████▊
| 14742/35912 [00:51<01:22, 256.11it/s]
 41%|█████████████████████████████▊
| 14775/35912 [00:51<01:17, 271.36it/s]
 41%|█████████████████████████████▉
| 14805/35912 [00:51<01:15, 278.80it/s]
 41%|█████████████████████████████▉
| 14834/35912 [00:51<01:15, 277.44it/s]
 41%|██████████████████████████████
| 14868/35912 [00:51<01:11, 292.34it/s]
 41%|██████████████████████████████
| 14901/35912 [00:51<01:09, 301.27it/s]
 42%|██████████████████████████████▏
```

```
| 14932/35912 [00:51<01:11, 292.90it/s]
 42%|████████████████▏
| 14965/35912 [00:51<01:09, 300.86it/s]
 42%|████████████████▏
| 14998/35912 [00:52<01:08, 307.56it/s]
 42%|████████████████▎
| 15030/35912 [00:52<01:07, 308.73it/s]
 42%|████████████████▎
| 15062/35912 [00:52<01:08, 303.41it/s]
 42%|████████████████▍
| 15096/35912 [00:52<01:06, 312.90it/s]
 42%|████████████████▍
| 15128/35912 [00:52<01:08, 305.35it/s]
 42%|████████████████▌
| 15159/35912 [00:52<01:13, 281.15it/s]
 42%|████████████████▌
| 15193/35912 [00:52<01:10, 295.21it/s]
 42%|████████████████▋
| 15224/35912 [00:52<01:12, 284.11it/s]
 42%|████████████████▋
| 15254/35912 [00:52<01:12, 285.63it/s]
 43%|████████████████▊
| 15286/35912 [00:52<01:10, 294.55it/s]
 43%|████████████████▊
| 15316/35912 [00:53<01:09, 295.52it/s]
 43%|█████████████████
| 15346/35912 [00:53<01:09, 295.31it/s]
 43%|█████████████████
| 15376/35912 [00:53<01:09, 294.32it/s]
 43%|█████████████████
| 15411/35912 [00:53<01:07, 304.48it/s]
 43%|█████████████████
| 15442/35912 [00:53<01:09, 295.01it/s]
 43%|█████████████████▏
| 15472/35912 [00:53<01:11, 284.93it/s]
 43%|█████████████████▎
| 15503/35912 [00:53<01:10, 290.59it/s]
 43%|█████████████████▎
| 15533/35912 [00:53<01:10, 287.69it/s]
 43%|█████████████████▍
| 15563/35912 [00:53<01:10, 290.65it/s]
 43%|█████████████████▍
| 15593/35912 [00:54<01:12, 282.06it/s]
 44%|█████████████████▌
| 15622/35912 [00:54<01:12, 278.09it/s]
 44%|█████████████████▌
| 15654/35912 [00:54<01:10, 286.57it/s]
 44%|█████████████████▋
| 15683/35912 [00:54<01:11, 284.44it/s]
 44%|█████████████████▋
| 15712/35912 [00:54<01:14, 271.09it/s]
 44%|█████████████████▋
| 15741/35912 [00:54<01:13, 275.93it/s]
 44%|█████████████████▊
| 15770/35912 [00:54<01:12, 278.60it/s]
 44%|█████████████████▉
| 15804/35912 [00:54<01:08, 293.25it/s]
```

```
44%|
| 15837/35912 [00:54<01:06, 302.77it/s]
44%|
| 15872/35912 [00:54<01:03, 314.93it/s]
44%|
| 15904/35912 [00:55<01:08, 291.64it/s]
44%|
| 15934/35912 [00:55<01:08, 290.91it/s]
44%|
| 15966/35912 [00:55<01:07, 297.63it/s]
45%|
| 15997/35912 [00:55<01:08, 292.12it/s]
45%|
| 16028/35912 [00:55<01:07, 293.28it/s]
45%|
| 16058/35912 [00:55<01:09, 285.40it/s]
45%|
| 16089/35912 [00:55<01:07, 291.76it/s]
45%|
| 16123/35912 [00:55<01:05, 303.29it/s]
45%|
| 16155/35912 [00:55<01:04, 306.60it/s]
45%|
| 16186/35912 [00:56<01:05, 301.59it/s]
45%|
| 16217/35912 [00:56<01:08, 286.62it/s]
45%|
| 16246/35912 [00:56<01:10, 278.75it/s]
45%|
| 16278/35912 [00:56<01:08, 288.61it/s]
45%|
| 16308/35912 [00:56<01:11, 276.07it/s]
46%|
| 16340/35912 [00:56<01:08, 286.59it/s]
46%|
| 16376/35912 [00:56<01:04, 302.40it/s]
46%|
| 16407/35912 [00:56<01:07, 287.14it/s]
46%|
| 16437/35912 [00:56<01:07, 288.59it/s]
46%|
| 16467/35912 [00:57<01:12, 267.95it/s]
46%|
| 16495/35912 [00:57<01:12, 269.32it/s]
46%|
| 16527/35912 [00:57<01:08, 282.21it/s]
46%|
| 16562/35912 [00:57<01:05, 296.78it/s]
46%|
| 16593/35912 [00:57<01:05, 294.03it/s]
46%|
| 16625/35912 [00:57<01:04, 299.90it/s]
46%|
| 16657/35912 [00:57<01:03, 302.44it/s]
46%|
| 16688/35912 [00:57<01:06, 289.57it/s]
47%|
```

```
| 16718/35912 [00:57<01:07, 283.73it/s]
47%|
| 16747/35912 [00:58<01:09, 275.26it/s]
47%|
| 16775/35912 [00:58<01:11, 266.63it/s]
47%|
| 16807/35912 [00:58<01:08, 277.21it/s]
47%|
| 16836/35912 [00:58<01:08, 279.54it/s]
47%|
| 16865/35912 [00:58<01:10, 270.21it/s]
47%|
| 16893/35912 [00:58<01:11, 264.78it/s]
47%|
| 16924/35912 [00:58<01:08, 275.62it/s]
47%|
| 16952/35912 [00:58<01:08, 275.51it/s]
47%|
| 16980/35912 [00:58<01:08, 275.43it/s]
47%|
| 17012/35912 [00:58<01:05, 286.87it/s]
47%|
| 17044/35912 [00:59<01:03, 295.47it/s]
48%|
| 17077/35912 [00:59<01:02, 303.59it/s]
48%|
| 17108/35912 [00:59<01:02, 298.68it/s]
48%|
| 17139/35912 [00:59<01:04, 292.00it/s]
48%|
| 17169/35912 [00:59<01:03, 293.72it/s]
48%|
| 17199/35912 [00:59<01:04, 291.51it/s]
48%|
| 17229/35912 [00:59<01:05, 285.04it/s]
48%|
| 17260/35912 [00:59<01:03, 291.49it/s]
48%|
| 17293/35912 [00:59<01:02, 299.82it/s]
48%|
| 17324/35912 [01:00<01:02, 297.82it/s]
48%|
| 17354/35912 [01:00<01:02, 295.19it/s]
48%|
| 17388/35912 [01:00<01:00, 305.91it/s]
49%|
| 17419/35912 [01:00<01:00, 303.76it/s]
49%|
| 17450/35912 [01:00<01:02, 297.08it/s]
49%|
| 17480/35912 [01:00<01:06, 278.32it/s]
49%|
| 17509/35912 [01:00<01:06, 277.11it/s]
49%|
| 17538/35912 [01:00<01:05, 278.65it/s]
49%|
| 17567/35912 [01:00<01:07, 271.89it/s]
```

```
 49%|████████████████████████████████████▍
| 17596/35912 [01:01<01:06, 274.93it/s]
 49%|████████████████████████████████████▌
| 17625/35912 [01:01<01:05, 278.70it/s]
 49%|████████████████████████████████████▌
| 17653/35912 [01:01<01:07, 270.43it/s]
 49%|████████████████████████████████████▋
| 17684/35912 [01:01<01:05, 279.13it/s]
 49%|████████████████████████████████████▋
| 17713/35912 [01:01<01:04, 280.05it/s]
 49%|█████████████████████████████████████
| 17743/35912 [01:01<01:03, 285.18it/s]
 50%|█████████████████████████████████████▏
| 17777/35912 [01:01<01:00, 299.10it/s]
 50%|█████████████████████████████████████▏
| 17808/35912 [01:01<01:00, 298.17it/s]
 50%|█████████████████████████████████████▎
| 17838/35912 [01:01<01:01, 294.56it/s]
 50%|█████████████████████████████████████▎
| 17868/35912 [01:01<01:02, 287.90it/s]
 50%|█████████████████████████████████████▍
| 17897/35912 [01:02<01:04, 277.99it/s]
 50%|█████████████████████████████████████▍
| 17929/35912 [01:02<01:02, 288.81it/s]
 50%|█████████████████████████████████████▌
| 17959/35912 [01:02<01:03, 283.23it/s]
 50%|█████████████████████████████████████▌
| 17988/35912 [01:02<01:03, 283.78it/s]
 50%|█████████████████████████████████████▊
| 18017/35912 [01:02<01:06, 269.17it/s]
 50%|█████████████████████████████████████▊
| 18051/35912 [01:02<01:02, 285.15it/s]
 50%|█████████████████████████████████████▉
| 18087/35912 [01:02<00:59, 300.52it/s]
 50%|█████████████████████████████████████▉
| 18118/35912 [01:02<01:01, 288.34it/s]
 51%|██████████████████████████████████████
| 18148/35912 [01:02<01:01, 291.12it/s]
 51%|██████████████████████████████████████
| 18181/35912 [01:03<00:59, 298.73it/s]
 51%|██████████████████████████████████████
| 18212/35912 [01:03<00:58, 300.51it/s]
 51%|██████████████████████████████████████
| 18245/35912 [01:03<00:57, 306.44it/s]
 51%|██████████████████████████████████████▏
| 18276/35912 [01:03<00:58, 301.48it/s]
 51%|██████████████████████████████████████▏
| 18307/35912 [01:03<01:01, 288.15it/s]
 51%|██████████████████████████████████████▎
| 18337/35912 [01:03<01:03, 275.78it/s]
 51%|██████████████████████████████████████▎
| 18371/35912 [01:03<01:00, 291.79it/s]
 51%|██████████████████████████████████████▍
| 18401/35912 [01:03<00:59, 292.72it/s]
 51%|██████████████████████████████████████▍
| 18431/35912 [01:03<01:00, 290.82it/s]
 51%|██████████████████████████████████████▌
```

```
| 18464/35912 [01:03<00:57, 300.93it/s]
 52%|████████████████████████████████▊
| 18495/35912 [01:04<01:02, 280.80it/s]
 52%|████████████████████████████████▊
| 18526/35912 [01:04<01:00, 288.38it/s]
 52%|████████████████████████████████▉
| 18556/35912 [01:04<01:01, 282.14it/s]
 52%|████████████████████████████████▉
| 18588/35912 [01:04<00:59, 291.14it/s]
 52%|████████████████████████████████▉
| 18620/35912 [01:04<00:58, 297.79it/s]
 52%|████████████████████████████████▉
| 18652/35912 [01:04<00:57, 302.63it/s]
 52%|█████████████████████████████████
| 18683/35912 [01:04<00:58, 294.64it/s]
 52%|█████████████████████████████████
| 18718/35912 [01:04<00:55, 307.10it/s]
 52%|█████████████████████████████████▏
| 18749/35912 [01:04<00:58, 295.89it/s]
 52%|█████████████████████████████████▏
| 18779/35912 [01:05<01:01, 277.59it/s]
 52%|█████████████████████████████████▎
| 18808/35912 [01:05<01:00, 280.60it/s]
 52%|█████████████████████████████████▎
| 18837/35912 [01:05<01:01, 277.09it/s]
 53%|█████████████████████████████████▎
| 18865/35912 [01:05<01:01, 276.54it/s]
 53%|█████████████████████████████████▎
| 18893/35912 [01:05<01:02, 272.93it/s]
 53%|█████████████████████████████████▍
| 18930/35912 [01:05<00:58, 290.17it/s]
 53%|█████████████████████████████████▍
| 18960/35912 [01:05<00:59, 284.95it/s]
 53%|█████████████████████████████████▌
| 18989/35912 [01:05<01:00, 281.67it/s]
 53%|█████████████████████████████████▌
| 19018/35912 [01:05<01:02, 272.36it/s]
 53%|█████████████████████████████████▌
| 19048/35912 [01:06<01:00, 278.75it/s]
 53%|█████████████████████████████████▌
| 19079/35912 [01:06<00:58, 286.07it/s]
 53%|█████████████████████████████████▋
| 19111/35912 [01:06<00:57, 290.07it/s]
 53%|█████████████████████████████████▋
| 19141/35912 [01:06<00:59, 280.89it/s]
 53%|██████████████████████████████████
| 19170/35912 [01:06<01:02, 268.83it/s]
 53%|██████████████████████████████████
| 19198/35912 [01:06<01:02, 267.62it/s]
 54%|██████████████████████████████████▏
| 19226/35912 [01:06<01:01, 270.64it/s]
 54%|██████████████████████████████████▏
| 19258/35912 [01:06<00:59, 280.26it/s]
 54%|██████████████████████████████████▎
| 19287/35912 [01:06<00:59, 279.25it/s]
 54%|██████████████████████████████████▎
| 19317/35912 [01:06<00:58, 283.77it/s]
```

```
 54%|█████████████████████▍
| 19346/35912 [01:07<00:58, 280.87it/s]
 54%|█████████████████████▍
| 19375/35912 [01:07<00:58, 281.30it/s]
 54%|█████████████████████▌
| 19404/35912 [01:07<00:58, 280.78it/s]
 54%|█████████████████████▌
| 19433/35912 [01:07<01:00, 271.02it/s]
 54%|█████████████████████▋
| 19461/35912 [01:07<01:01, 266.84it/s]
 54%|█████████████████████▋
| 19493/35912 [01:07<00:58, 278.83it/s]
 54%|█████████████████████▋
| 19526/35912 [01:07<00:56, 291.87it/s]
 54%|█████████████████████▋
| 19561/35912 [01:07<00:54, 302.62it/s]
 55%|█████████████████████▊
| 19594/35912 [01:07<00:52, 309.70it/s]
 55%|█████████████████████▊
| 19626/35912 [01:08<00:55, 295.66it/s]
 55%|█████████████████████▉
| 19659/35912 [01:08<00:53, 302.90it/s]
 55%|█████████████████████▉
| 19691/35912 [01:08<00:52, 306.30it/s]
 55%|█████████████████████▉
| 19722/35912 [01:08<00:55, 289.59it/s]
 55%|██████████████████████
| 19752/35912 [01:08<00:56, 285.37it/s]
 55%|██████████████████████
| 19782/35912 [01:08<00:56, 286.52it/s]
 55%|██████████████████████
| 19811/35912 [01:08<00:58, 275.51it/s]
 55%|██████████████████████
| 19846/35912 [01:08<00:55, 291.56it/s]
 55%|██████████████████████▏
| 19876/35912 [01:08<00:54, 291.71it/s]
 55%|██████████████████████▏
| 19907/35912 [01:09<00:54, 293.82it/s]
 56%|██████████████████████▏
| 19941/35912 [01:09<00:52, 305.69it/s]
 56%|██████████████████████▏
| 19972/35912 [01:09<00:52, 302.74it/s]
 56%|██████████████████████▎
| 20003/35912 [01:09<00:56, 283.42it/s]
 56%|██████████████████████▎
| 20032/35912 [01:09<01:01, 258.89it/s]
 56%|██████████████████████▎
| 20066/35912 [01:09<00:56, 278.34it/s]
 56%|██████████████████████▎
| 20099/35912 [01:09<00:54, 289.17it/s]
 56%|██████████████████████▍
| 20129/35912 [01:09<00:55, 285.11it/s]
 56%|██████████████████████▍
| 20160/35912 [01:09<00:54, 291.54it/s]
 56%|██████████████████████▍
| 20196/35912 [01:09<00:51, 307.81it/s]
 56%|██████████████████████▌
```

```
| 20228/35912 [01:10<00:54, 285.80it/s]
56%|
| 20258/35912 [01:10<00:54, 287.64it/s]
56%|
| 20290/35912 [01:10<00:52, 296.04it/s]
57%|
| 20321/35912 [01:10<00:52, 297.74it/s]
57%|
| 20352/35912 [01:10<00:52, 298.94it/s]
57%|
| 20383/35912 [01:10<00:52, 294.66it/s]
57%|
| 20418/35912 [01:10<00:50, 307.93it/s]
57%|
| 20450/35912 [01:10<00:50, 303.73it/s]
57%|
| 20481/35912 [01:10<00:51, 299.64it/s]
57%|
| 20514/35912 [01:11<00:50, 306.66it/s]
57%|
| 20545/35912 [01:11<00:50, 301.62it/s]
57%|
| 20578/35912 [01:11<00:49, 308.97it/s]
57%|
| 20610/35912 [01:11<00:49, 307.95it/s]
57%|
| 20641/35912 [01:11<00:51, 295.59it/s]
58%|
| 20671/35912 [01:11<00:52, 291.10it/s]
58%|
| 20701/35912 [01:11<00:54, 281.56it/s]
58%|
| 20730/35912 [01:11<00:54, 280.16it/s]
58%|
| 20759/35912 [01:11<00:56, 269.10it/s]
58%|
| 20791/35912 [01:12<00:54, 279.10it/s]
58%|
| 20826/35912 [01:12<00:51, 290.71it/s]
58%|
| 20856/35912 [01:12<00:53, 280.52it/s]
58%|
| 20887/35912 [01:12<00:52, 288.17it/s]
58%|
| 20921/35912 [01:12<00:49, 301.39it/s]
58%|
| 20954/35912 [01:12<00:48, 307.94it/s]
58%|
| 20986/35912 [01:12<00:52, 284.36it/s]
59%|
| 21017/35912 [01:12<00:51, 290.99it/s]
59%|
| 21048/35912 [01:12<00:50, 294.15it/s]
59%|
| 21079/35912 [01:13<00:49, 298.10it/s]
59%|
| 21111/35912 [01:13<00:48, 303.71it/s]
```

```
 59%|
| 21142/35912 [01:13<00:49, 297.90it/s]
 59%|
| 21172/35912 [01:13<00:49, 297.87it/s]
 59%|
| 21202/35912 [01:13<00:50, 290.11it/s]
 59%|
| 21232/35912 [01:13<00:51, 286.53it/s]
 59%|
| 21267/35912 [01:13<00:48, 301.66it/s]
 59%|
| 21298/35912 [01:13<00:49, 294.83it/s]
 59%|
| 21328/35912 [01:13<00:50, 286.43it/s]
 59%|
| 21363/35912 [01:13<00:48, 301.58it/s]
 60%|
| 21394/35912 [01:14<00:47, 302.52it/s]
 60%|
| 21425/35912 [01:14<00:50, 288.81it/s]
 60%|
| 21455/35912 [01:14<00:49, 291.46it/s]
 60%|
| 21487/35912 [01:14<00:48, 296.37it/s]
 60%|
| 21517/35912 [01:14<00:52, 273.34it/s]
 60%|
| 21547/35912 [01:14<00:51, 280.25it/s]
 60%|
| 21576/35912 [01:14<00:51, 276.07it/s]
 60%|
| 21606/35912 [01:14<00:50, 281.46it/s]
 60%|
| 21635/35912 [01:14<00:50, 283.36it/s]
 60%|
| 21665/35912 [01:15<00:50, 281.89it/s]
 60%|
| 21694/35912 [01:15<00:50, 282.84it/s]
 60%|
| 21724/35912 [01:15<00:49, 284.73it/s]
 61%|
| 21755/35912 [01:15<00:48, 290.45it/s]
 61%|
| 21787/35912 [01:15<00:47, 298.11it/s]
 61%|
| 21820/35912 [01:15<00:46, 304.70it/s]
 61%|
| 21851/35912 [01:15<00:47, 296.85it/s]
 61%|
| 21885/35912 [01:15<00:46, 303.06it/s]
 61%|
| 21918/35912 [01:15<00:45, 310.03it/s]
 61%|
| 21950/35912 [01:15<00:44, 312.28it/s]
 61%|
| 21983/35912 [01:16<00:44, 309.62it/s]
 61%|
```

```
| 22015/35912 [01:16<00:46, 298.08it/s]
 61%|██████████████████████████████████████████
| 22045/35912 [01:16<00:46, 295.37it/s]
 61%|██████████████████████████████████████████
| 22077/35912 [01:16<00:45, 300.88it/s]
 62%|██████████████████████████████████████████▏
| 22108/35912 [01:16<00:48, 285.39it/s]
 62%|██████████████████████████████████████████▏
| 22139/35912 [01:16<00:47, 289.30it/s]
 62%|██████████████████████████████████████████▎
| 22169/35912 [01:16<00:48, 282.76it/s]
 62%|██████████████████████████████████████████▎
| 22201/35912 [01:16<00:47, 290.81it/s]
 62%|██████████████████████████████████████████▍
| 22231/35912 [01:16<00:48, 281.37it/s]
 62%|██████████████████████████████████████████▍
| 22261/35912 [01:17<00:47, 284.49it/s]
 62%|███████████████████████████████████████████
| 22291/35912 [01:17<00:47, 287.54it/s]
 62%|███████████████████████████████████████████
| 22321/35912 [01:17<00:46, 290.55it/s]
 62%|██████████████████████████████████████████▋
| 22356/35912 [01:17<00:44, 303.97it/s]
 62%|██████████████████████████████████████████▊
| 22387/35912 [01:17<00:47, 284.18it/s]
 62%|██████████████████████████████████████████▊
| 22419/35912 [01:17<00:45, 293.46it/s]
 63%|██████████████████████████████████████████▉
| 22450/35912 [01:17<00:45, 296.76it/s]
 63%|██████████████████████████████████████████▉
| 22481/35912 [01:17<00:44, 299.10it/s]
 63%|███████████████████████████████████████████
| 22512/35912 [01:17<00:47, 285.05it/s]
 63%|███████████████████████████████████████████
| 22543/35912 [01:18<00:46, 289.87it/s]
 63%|███████████████████████████████████████████▏
| 22576/35912 [01:18<00:44, 298.62it/s]
 63%|███████████████████████████████████████████▏
| 22608/35912 [01:18<00:44, 301.52it/s]
 63%|███████████████████████████████████████████▎
| 22639/35912 [01:18<00:46, 287.35it/s]
 63%|███████████████████████████████████████████▎
| 22668/35912 [01:18<00:46, 287.54it/s]
 63%|███████████████████████████████████████████▍
| 22700/35912 [01:18<00:44, 294.33it/s]
 63%|███████████████████████████████████████████▍
| 22731/35912 [01:18<00:44, 297.38it/s]
 63%|████████████████████████████████████████████
| 22761/35912 [01:18<00:45, 287.28it/s]
 63%|████████████████████████████████████████████
| 22795/35912 [01:18<00:43, 299.13it/s]
 64%|███████████████████████████████████████████▋
| 22826/35912 [01:18<00:46, 279.68it/s]
 64%|███████████████████████████████████████████▋
| 22857/35912 [01:19<00:45, 285.17it/s]
 64%|███████████████████████████████████████████▊
| 22886/35912 [01:19<00:47, 276.21it/s]
```

```
64%|████████████████████████████▋
| 22915/35912 [01:19<00:46, 279.61it/s]
64%|████████████████████████████▊
| 22945/35912 [01:19<00:45, 284.03it/s]
64%|████████████████████████████▊
| 22975/35912 [01:19<00:44, 288.03it/s]
64%|█████████████████████████████
| 23004/35912 [01:19<00:45, 285.45it/s]
64%|█████████████████████████████
| 23033/35912 [01:19<00:45, 283.67it/s]
64%|█████████████████████████████▏
| 23070/35912 [01:19<00:42, 303.72it/s]
64%|█████████████████████████████▏
| 23101/35912 [01:19<00:43, 296.20it/s]
64%|█████████████████████████████▎
| 23131/35912 [01:20<00:43, 294.94it/s]
64%|█████████████████████████████▎
| 23161/35912 [01:20<00:45, 281.67it/s]
65%|█████████████████████████████▍
| 23195/35912 [01:20<00:43, 294.09it/s]
65%|█████████████████████████████▍
| 23226/35912 [01:20<00:42, 296.35it/s]
65%|█████████████████████████████▍
| 23256/35912 [01:20<00:45, 280.20it/s]
65%|█████████████████████████████▋
| 23289/35912 [01:20<00:43, 291.37it/s]
65%|█████████████████████████████▋
| 23319/35912 [01:20<00:42, 293.28it/s]
65%|█████████████████████████████▊
| 23349/35912 [01:20<00:42, 293.76it/s]
65%|█████████████████████████████▊
| 23379/35912 [01:20<00:42, 294.11it/s]
65%|█████████████████████████████▉
| 23409/35912 [01:20<00:42, 295.21it/s]
65%|█████████████████████████████▉
| 23439/35912 [01:21<00:43, 285.05it/s]
65%|█████████████████████████████▋
| 23470/35912 [01:21<00:42, 290.68it/s]
65%|█████████████████████████████▋
| 23500/35912 [01:21<00:42, 290.25it/s]
66%|█████████████████████████████▊
| 23530/35912 [01:21<00:42, 289.11it/s]
66%|█████████████████████████████▊
| 23561/35912 [01:21<00:42, 293.62it/s]
66%|█████████████████████████████▉
| 23592/35912 [01:21<00:41, 296.02it/s]
66%|█████████████████████████████▉
| 23622/35912 [01:21<00:41, 294.82it/s]
66%|██████████████████████████████
| 23653/35912 [01:21<00:41, 297.72it/s]
66%|██████████████████████████████
| 23683/35912 [01:21<00:41, 294.26it/s]
66%|██████████████████████████████
| 23715/35912 [01:22<00:40, 300.91it/s]
66%|██████████████████████████████
| 23746/35912 [01:22<00:40, 297.69it/s]
66%|██████████████████████████████▍
```

```
| 23776/35912 [01:22<00:41, 290.84it/s]
 66%|
| 23806/35912 [01:22<00:44, 274.46it/s]
 66%|
| 23835/35912 [01:22<00:43, 276.77it/s]
 66%|
| 23863/35912 [01:22<00:43, 274.69it/s]
 67%|
| 23891/35912 [01:22<00:43, 274.05it/s]
 67%|
| 23922/35912 [01:22<00:42, 283.36it/s]
 67%|
| 23951/35912 [01:22<00:42, 284.70it/s]
 67%|
| 23981/35912 [01:22<00:41, 287.69it/s]
 67%|
| 24010/35912 [01:23<00:42, 280.26it/s]
 67%|
| 24040/35912 [01:23<00:41, 282.90it/s]
 67%|
| 24074/35912 [01:23<00:39, 296.55it/s]
 67%|
| 24104/35912 [01:23<00:40, 293.46it/s]
 67%|
| 24134/35912 [01:23<00:39, 294.76it/s]
 67%|
| 24164/35912 [01:23<00:39, 293.94it/s]
 67%|
| 24194/35912 [01:23<00:40, 292.51it/s]
 67%|
| 24227/35912 [01:23<00:38, 300.58it/s]
 68%|
| 24258/35912 [01:23<00:38, 300.06it/s]
 68%|
| 24289/35912 [01:24<00:39, 292.10it/s]
 68%|
| 24322/35912 [01:24<00:38, 299.46it/s]
 68%|
| 24355/35912 [01:24<00:37, 307.38it/s]
 68%|
| 24387/35912 [01:24<00:37, 305.96it/s]
 68%|
| 24418/35912 [01:24<00:37, 304.66it/s]
 68%|
| 24451/35912 [01:24<00:36, 311.23it/s]
 68%|
| 24483/35912 [01:24<00:38, 294.19it/s]
 68%|
| 24513/35912 [01:24<00:38, 295.27it/s]
 68%|
| 24543/35912 [01:24<00:38, 291.71it/s]
 68%|
| 24577/35912 [01:24<00:37, 303.30it/s]
 69%|
| 24608/35912 [01:25<00:38, 290.93it/s]
 69%|
| 24638/35912 [01:25<00:38, 291.27it/s]
```

```
69%|██████████████████████████████████████████████████████████████████████
| 24668/35912 [01:25<00:39, 286.50it/s]
69%|██████████████████████████████████████████████████████████████████████
| 24698/35912 [01:25<00:39, 286.49it/s]
69%|██████████████████████████████████████████████████████████████████████▏
| 24727/35912 [01:25<00:41, 271.64it/s]
69%|██████████████████████████████████████████████████████████████████████▏
| 24760/35912 [01:25<00:39, 285.57it/s]
69%|██████████████████████████████████████████████████████████████████████▎
| 24791/35912 [01:25<00:38, 291.06it/s]
69%|██████████████████████████████████████████████████████████████████████▎
| 24821/35912 [01:25<00:38, 288.84it/s]
69%|██████████████████████████████████████████████████████████████████████▍
| 24851/35912 [01:25<00:39, 278.52it/s]
69%|██████████████████████████████████████████████████████████████████████▍
| 24880/35912 [01:26<00:39, 281.27it/s]
69%|███████████████████████████████████████████████████████████████████████
| 24911/35912 [01:26<00:38, 288.72it/s]
69%|███████████████████████████████████████████████████████████████████████
| 24946/35912 [01:26<00:36, 300.24it/s]
70%|███████████████████████████████████████████████████████████████████████▏
| 24977/35912 [01:26<00:37, 292.22it/s]
70%|███████████████████████████████████████████████████████████████████████▏
| 25007/35912 [01:26<00:37, 289.63it/s]
70%|███████████████████████████████████████████████████████████████████████▏
| 25038/35912 [01:26<00:36, 294.84it/s]
70%|███████████████████████████████████████████████████████████████████████▏
| 25071/35912 [01:26<00:35, 303.96it/s]
70%|███████████████████████████████████████████████████████████████████████▎
| 25102/35912 [01:26<00:35, 301.54it/s]
70%|███████████████████████████████████████████████████████████████████████▎
| 25134/35912 [01:26<00:35, 305.33it/s]
70%|████████████████████████████████████████████████████████████████████████
| 25165/35912 [01:26<00:35, 298.99it/s]
70%|████████████████████████████████████████████████████████████████████████
| 25196/35912 [01:27<00:35, 301.56it/s]
70%|████████████████████████████████████████████████████████████████████████▍
| 25227/35912 [01:27<00:36, 293.09it/s]
70%|████████████████████████████████████████████████████████████████████████▍
| 25257/35912 [01:27<00:36, 291.93it/s]
70%|████████████████████████████████████████████████████████████████████████▌
| 25288/35912 [01:27<00:36, 294.81it/s]
71%|████████████████████████████████████████████████████████████████████████▌
| 25318/35912 [01:27<00:36, 293.12it/s]
71%|████████████████████████████████████████████████████████████████████████▌
| 25348/35912 [01:27<00:37, 282.88it/s]
71%|████████████████████████████████████████████████████████████████████████▌
| 25377/35912 [01:27<00:37, 282.71it/s]
71%|█████████████████████████████████████████████████████████████████████████
| 25410/35912 [01:27<00:35, 294.04it/s]
71%|█████████████████████████████████████████████████████████████████████████▏
| 25440/35912 [01:27<00:35, 295.17it/s]
71%|█████████████████████████████████████████████████████████████████████████▏
| 25470/35912 [01:28<00:36, 286.65it/s]
71%|█████████████████████████████████████████████████████████████████████████▏
| 25499/35912 [01:28<00:36, 282.81it/s]
71%|█████████████████████████████████████████████████████████████████████████▏
```

```
| 25528/35912 [01:28<00:36, 282.68it/s]
 71%|█████████████████████████████████████████████████████▏
| 25557/35912 [01:28<00:36, 284.23it/s]
 71%|█████████████████████████████████████████████████████▎
| 25586/35912 [01:28<00:37, 277.95it/s]
 71%|█████████████████████████████████████████████████████▎
| 25614/35912 [01:28<00:38, 269.94it/s]
 71%|█████████████████████████████████████████████████████▌
| 25644/35912 [01:28<00:36, 277.74it/s]
 72%|█████████████████████████████████████████████████████▋
| 25678/35912 [01:28<00:35, 289.59it/s]
 72%|█████████████████████████████████████████████████████▋
| 25712/35912 [01:28<00:33, 300.09it/s]
 72%|█████████████████████████████████████████████████████▊
| 25745/35912 [01:28<00:33, 306.99it/s]
 72%|█████████████████████████████████████████████████████▊
| 25776/35912 [01:29<00:33, 298.36it/s]
 72%|█████████████████████████████████████████████████████▉
| 25807/35912 [01:29<00:33, 300.25it/s]
 72%|█████████████████████████████████████████████████████▉
| 25840/35912 [01:29<00:32, 306.26it/s]
 72%|██████████████████████████████████████████████████████
| 25871/35912 [01:29<00:34, 291.17it/s]
 72%|██████████████████████████████████████████████████████
| 25901/35912 [01:29<00:35, 282.42it/s]
 72%|██████████████████████████████████████████████████████▏
| 25934/35912 [01:29<00:34, 293.05it/s]
 72%|██████████████████████████████████████████████████████▏
| 25964/35912 [01:29<00:34, 290.20it/s]
 72%|██████████████████████████████████████████████████████▎
| 25995/35912 [01:29<00:33, 294.41it/s]
 72%|██████████████████████████████████████████████████████▎
| 26029/35912 [01:29<00:32, 306.15it/s]
 73%|██████████████████████████████████████████████████████▍
| 26060/35912 [01:30<00:33, 291.94it/s]
 73%|██████████████████████████████████████████████████████▍
| 26090/35912 [01:30<00:34, 285.32it/s]
 73%|██████████████████████████████████████████████████████▌
| 26119/35912 [01:30<00:35, 278.69it/s]
 73%|██████████████████████████████████████████████████████▌
| 26151/35912 [01:30<00:33, 288.56it/s]
 73%|██████████████████████████████████████████████████████▋
| 26186/35912 [01:30<00:32, 303.23it/s]
 73%|██████████████████████████████████████████████████████▊
| 26217/35912 [01:30<00:33, 289.25it/s]
 73%|██████████████████████████████████████████████████████▊
| 26247/35912 [01:30<00:35, 272.74it/s]
 73%|██████████████████████████████████████████████████████▉
| 26279/35912 [01:30<00:33, 284.07it/s]
 73%|██████████████████████████████████████████████████████▉
| 26309/35912 [01:30<00:33, 284.77it/s]
 73%|███████████████████████████████████████████████████████
| 26340/35912 [01:31<00:32, 291.32it/s]
 73%|███████████████████████████████████████████████████████
| 26370/35912 [01:31<00:32, 292.38it/s]
 74%|███████████████████████████████████████████████████████▏
| 26402/35912 [01:31<00:31, 299.51it/s]
```

```
 74%|█████████████████████████████████████▍
| 26438/35912 [01:31<00:30, 314.01it/s]
 74%|█████████████████████████████████████▍
| 26470/35912 [01:31<00:30, 306.08it/s]
 74%|█████████████████████████████████████▍
| 26501/35912 [01:31<00:31, 296.92it/s]
 74%|█████████████████████████████████████▌
| 26531/35912 [01:31<00:33, 282.14it/s]
 74%|█████████████████████████████████████▌
| 26563/35912 [01:31<00:32, 291.14it/s]
 74%|█████████████████████████████████████▌
| 26593/35912 [01:31<00:31, 291.41it/s]
 74%|█████████████████████████████████████▌
| 26628/35912 [01:31<00:30, 306.23it/s]
 74%|█████████████████████████████████████▋
| 26659/35912 [01:32<00:30, 299.56it/s]
 74%|█████████████████████████████████████▋
| 26690/35912 [01:32<00:31, 295.97it/s]
 74%|█████████████████████████████████████▊
| 26720/35912 [01:32<00:32, 283.12it/s]
 74%|█████████████████████████████████████▊
| 26751/35912 [01:32<00:31, 288.47it/s]
 75%|█████████████████████████████████████▊
| 26781/35912 [01:32<00:33, 271.50it/s]
 75%|█████████████████████████████████████▊
| 26814/35912 [01:32<00:32, 283.27it/s]
 75%|██████████████████████████████████████
| 26845/35912 [01:32<00:31, 290.19it/s]
 75%|██████████████████████████████████████▏
| 26879/35912 [01:32<00:30, 299.74it/s]
 75%|██████████████████████████████████████▏
| 26910/35912 [01:32<00:30, 298.62it/s]
 75%|██████████████████████████████████████▎
| 26941/35912 [01:33<00:32, 274.90it/s]
 75%|██████████████████████████████████████▎
| 26974/35912 [01:33<00:31, 286.59it/s]
 75%|██████████████████████████████████████▍
| 27004/35912 [01:33<00:31, 282.54it/s]
 75%|██████████████████████████████████████▍
| 27036/35912 [01:33<00:30, 292.24it/s]
 75%|██████████████████████████████████████
| 27066/35912 [01:33<00:32, 272.34it/s]
 75%|██████████████████████████████████████
| 27094/35912 [01:33<00:33, 265.45it/s]
 76%|██████████████████████████████████████▋
| 27121/35912 [01:33<00:33, 263.12it/s]
 76%|██████████████████████████████████████▋
| 27148/35912 [01:33<00:33, 261.51it/s]
 76%|██████████████████████████████████████▊
| 27177/35912 [01:33<00:32, 268.16it/s]
 76%|██████████████████████████████████████▊
| 27207/35912 [01:34<00:31, 276.42it/s]
 76%|██████████████████████████████████████▉
| 27237/35912 [01:34<00:30, 281.72it/s]
 76%|██████████████████████████████████████▉
| 27268/35912 [01:34<00:29, 289.05it/s]
 76%|███████████████████████████████████████
```

```
| 27299/35912 [01:34<00:29, 294.39it/s]
 76%|
| 27329/35912 [01:34<00:29, 293.71it/s]
 76%|
| 27359/35912 [01:34<00:30, 280.10it/s]
 76%|
| 27392/35912 [01:34<00:29, 291.29it/s]
 76%|
| 27422/35912 [01:34<00:29, 288.17it/s]
 76%|
| 27452/35912 [01:34<00:30, 281.20it/s]
 77%|
| 27481/35912 [01:35<00:30, 277.51it/s]
 77%|
| 27509/35912 [01:35<00:30, 272.79it/s]
 77%|
| 27544/35912 [01:35<00:28, 291.59it/s]
 77%|
| 27574/35912 [01:35<00:30, 277.21it/s]
 77%|
| 27604/35912 [01:35<00:29, 283.09it/s]
 77%|
| 27634/35912 [01:35<00:28, 286.53it/s]
 77%|
| 27666/35912 [01:35<00:27, 295.21it/s]
 77%|
| 27696/35912 [01:35<00:28, 291.69it/s]
 77%|
| 27726/35912 [01:35<00:28, 282.74it/s]
 77%|
| 27756/35912 [01:35<00:28, 287.10it/s]
 77%|
| 27785/35912 [01:36<00:29, 272.02it/s]
 77%|
| 27815/35912 [01:36<00:28, 279.28it/s]
 78%|
| 27847/35912 [01:36<00:28, 285.15it/s]
 78%|
| 27876/35912 [01:36<00:28, 282.63it/s]
 78%|
| 27906/35912 [01:36<00:28, 284.58it/s]
 78%|
| 27938/35912 [01:36<00:27, 293.76it/s]
 78%|
| 27971/35912 [01:36<00:26, 303.16it/s]
 78%|
| 28002/35912 [01:36<00:25, 304.52it/s]
 78%|
| 28033/35912 [01:36<00:27, 291.72it/s]
 78%|
| 28063/35912 [01:37<00:27, 285.18it/s]
 78%|
| 28094/35912 [01:37<00:26, 291.59it/s]
 78%|
| 28124/35912 [01:37<00:26, 290.88it/s]
 78%|
| 28154/35912 [01:37<00:26, 291.23it/s]
```

```
 78%|
| 28186/35912 [01:37<00:26, 296.21it/s]
 79%|
| 28219/35912 [01:37<00:25, 303.30it/s]
 79%|
| 28250/35912 [01:37<00:25, 297.62it/s]
 79%|
| 28283/35912 [01:37<00:24, 306.02it/s]
 79%|
| 28314/35912 [01:37<00:26, 285.43it/s]
 79%|
| 28343/35912 [01:37<00:26, 282.00it/s]
 79%|
| 28376/35912 [01:38<00:25, 293.50it/s]
 79%|
| 28407/35912 [01:38<00:25, 297.64it/s]
 79%|
| 28437/35912 [01:38<00:26, 283.38it/s]
 79%|
| 28467/35912 [01:38<00:25, 287.58it/s]
 79%|
| 28496/35912 [01:38<00:27, 272.32it/s]
 79%|
| 28525/35912 [01:38<00:26, 273.68it/s]
 80%|
| 28559/35912 [01:38<00:25, 287.94it/s]
 80%|
| 28589/35912 [01:38<00:25, 290.00it/s]
 80%|
| 28619/35912 [01:38<00:24, 292.30it/s]
 80%|
| 28652/35912 [01:39<00:24, 301.24it/s]
 80%|
| 28683/35912 [01:39<00:23, 303.17it/s]
 80%|
| 28714/35912 [01:39<00:23, 304.53it/s]
 80%|
| 28745/35912 [01:39<00:24, 296.74it/s]
 80%|
| 28775/35912 [01:39<00:24, 287.69it/s]
 80%|
| 28806/35912 [01:39<00:24, 289.33it/s]
 80%|
| 28836/35912 [01:39<00:24, 287.64it/s]
 80%|
| 28865/35912 [01:39<00:24, 286.87it/s]
 80%|
| 28894/35912 [01:39<00:24, 284.64it/s]
 81%|
| 28927/35912 [01:39<00:23, 294.72it/s]
 81%|
| 28957/35912 [01:40<00:24, 287.99it/s]
 81%|
| 28988/35912 [01:40<00:23, 292.00it/s]
 81%|
| 29021/35912 [01:40<00:23, 299.38it/s]
 81%|
```

```
| 29052/35912 [01:40<00:23, 287.60it/s]
81%|████████████████████████████████████████████████████████████▍
| 29081/35912 [01:40<00:24, 281.02it/s]
81%|████████████████████████████████████████████████████████████▌
| 29110/35912 [01:40<00:24, 279.77it/s]
81%|████████████████████████████████████████████████████████████▌
| 29142/35912 [01:40<00:23, 288.60it/s]
81%|████████████████████████████████████████████████████████████▋
| 29172/35912 [01:40<00:23, 286.32it/s]
81%|████████████████████████████████████████████████████████████▋
| 29203/35912 [01:40<00:23, 289.97it/s]
81%|█████████████████████████████████████████████████████████████
| 29233/35912 [01:41<00:23, 280.82it/s]
81%|█████████████████████████████████████████████████████████████
| 29262/35912 [01:41<00:24, 275.67it/s]
82%|█████████████████████████████████████████████████████████████▏
| 29293/35912 [01:41<00:23, 284.57it/s]
82%|█████████████████████████████████████████████████████████████▏
| 29322/35912 [01:41<00:23, 284.71it/s]
82%|█████████████████████████████████████████████████████████████▎
| 29352/35912 [01:41<00:22, 286.88it/s]
82%|█████████████████████████████████████████████████████████████▎
| 29381/35912 [01:41<00:23, 280.53it/s]
82%|█████████████████████████████████████████████████████████████▍
| 29415/35912 [01:41<00:22, 293.22it/s]
82%|█████████████████████████████████████████████████████████████▍
| 29445/35912 [01:41<00:22, 292.86it/s]
82%|█████████████████████████████████████████████████████████████▌
| 29475/35912 [01:41<00:22, 289.24it/s]
82%|█████████████████████████████████████████████████████████████▌
| 29507/35912 [01:41<00:21, 293.95it/s]
82%|█████████████████████████████████████████████████████████████▋
| 29537/35912 [01:42<00:22, 288.32it/s]
82%|█████████████████████████████████████████████████████████████▋
| 29567/35912 [01:42<00:21, 289.42it/s]
82%|█████████████████████████████████████████████████████████████▊
| 29597/35912 [01:42<00:22, 282.04it/s]
83%|█████████████████████████████████████████████████████████████▉
| 29631/35912 [01:42<00:21, 295.90it/s]
83%|█████████████████████████████████████████████████████████████▉
| 29668/35912 [01:42<00:20, 311.85it/s]
83%|██████████████████████████████████████████████████████████████
| 29700/35912 [01:42<00:20, 303.78it/s]
83%|██████████████████████████████████████████████████████████████
| 29732/35912 [01:42<00:20, 306.94it/s]
83%|██████████████████████████████████████████████████████████████▏
| 29765/35912 [01:42<00:19, 310.22it/s]
83%|██████████████████████████████████████████████████████████████▏
| 29798/35912 [01:42<00:19, 311.68it/s]
83%|██████████████████████████████████████████████████████████████▎
| 29830/35912 [01:43<00:19, 313.45it/s]
83%|██████████████████████████████████████████████████████████████▎
| 29862/35912 [01:43<00:19, 314.69it/s]
83%|██████████████████████████████████████████████████████████████▍
| 29894/35912 [01:43<00:20, 286.82it/s]
83%|██████████████████████████████████████████████████████████████▍
| 29924/35912 [01:43<00:21, 277.98it/s]
```

```
83%|
| 29953/35912 [01:43<00:21, 279.26it/s]
83%|
| 29982/35912 [01:43<00:21, 276.18it/s]
84%|
| 30011/35912 [01:43<00:21, 278.79it/s]
84%|
| 30040/35912 [01:43<00:21, 274.29it/s]
84%|
| 30076/35912 [01:43<00:19, 294.12it/s]
84%|
| 30106/35912 [01:44<00:20, 277.28it/s]
84%|
| 30139/35912 [01:44<00:19, 290.65it/s]
84%|
| 30169/35912 [01:44<00:20, 286.93it/s]
84%|
| 30199/35912 [01:44<00:20, 277.25it/s]
84%|
| 30230/35912 [01:44<00:19, 284.99it/s]
84%|
| 30264/35912 [01:44<00:18, 298.95it/s]
84%|
| 30295/35912 [01:44<00:19, 288.91it/s]
84%|
| 30325/35912 [01:44<00:19, 286.54it/s]
85%|
| 30354/35912 [01:44<00:19, 280.31it/s]
85%|
| 30387/35912 [01:44<00:18, 292.22it/s]
85%|
| 30417/35912 [01:45<00:20, 270.86it/s]
85%|
| 30448/35912 [01:45<00:19, 279.45it/s]
85%|
| 30477/35912 [01:45<00:19, 278.69it/s]
85%|
| 30511/35912 [01:45<00:18, 293.31it/s]
85%|
| 30541/35912 [01:45<00:18, 286.24it/s]
85%|
| 30571/35912 [01:45<00:18, 285.49it/s]
85%|
| 30605/35912 [01:45<00:17, 298.55it/s]
85%|
| 30636/35912 [01:45<00:18, 280.08it/s]
85%|
| 30665/35912 [01:45<00:18, 281.56it/s]
85%|
| 30697/35912 [01:46<00:17, 291.50it/s]
86%|
| 30730/35912 [01:46<00:17, 296.61it/s]
86%|
| 30760/35912 [01:46<00:18, 285.96it/s]
86%|
| 30789/35912 [01:46<00:18, 282.35it/s]
86%|
```

```
| 30821/35912 [01:46<00:17, 290.52it/s]
 86%|
| 30851/35912 [01:46<00:17, 288.46it/s]
 86%|
| 30887/35912 [01:46<00:16, 304.63it/s]
 86%|
| 30918/35912 [01:46<00:17, 292.61it/s]
 86%|
| 30948/35912 [01:46<00:17, 290.75it/s]
 86%|
| 30978/35912 [01:47<00:17, 278.98it/s]
 86%|
| 31011/35912 [01:47<00:16, 291.20it/s]
 86%|
| 31041/35912 [01:47<00:17, 277.74it/s]
 87%|
| 31071/35912 [01:47<00:17, 277.98it/s]
 87%|
| 31100/35912 [01:47<00:17, 269.18it/s]
 87%|
| 31132/35912 [01:47<00:17, 278.43it/s]
 87%|
| 31161/35912 [01:47<00:16, 281.20it/s]
 87%|
| 31192/35912 [01:47<00:16, 283.93it/s]
 87%|
| 31221/35912 [01:47<00:17, 271.53it/s]
 87%|
| 31249/35912 [01:48<00:17, 267.95it/s]
 87%|
| 31276/35912 [01:48<00:17, 261.76it/s]
 87%|
| 31307/35912 [01:48<00:16, 274.04it/s]
 87%|
| 31336/35912 [01:48<00:16, 272.58it/s]
 87%|
| 31370/35912 [01:48<00:15, 289.28it/s]
 87%|
| 31401/35912 [01:48<00:15, 294.59it/s]
 88%|
| 31431/35912 [01:48<00:16, 279.10it/s]
 88%|
| 31462/35912 [01:48<00:15, 287.12it/s]
 88%|
| 31495/35912 [01:48<00:14, 298.17it/s]
 88%|
| 31526/35912 [01:48<00:14, 296.67it/s]
 88%|
| 31559/35912 [01:49<00:14, 303.64it/s]
 88%|
| 31590/35912 [01:49<00:14, 296.99it/s]
 88%|
| 31622/35912 [01:49<00:14, 302.03it/s]
 88%|
| 31653/35912 [01:49<00:14, 296.79it/s]
 88%|
| 31684/35912 [01:49<00:14, 299.13it/s]
```

```
88%|██████████████████████████████████████████████████████████████████▏
| 31715/35912 [01:49<00:14, 299.05it/s]
88%|██████████████████████████████████████████████████████████████████▎
| 31745/35912 [01:49<00:15, 273.43it/s]
88%|██████████████████████████████████████████████████████████████████▎
| 31777/35912 [01:49<00:14, 283.08it/s]
89%|██████████████████████████████████████████████████████████████████▍
| 31806/35912 [01:49<00:15, 273.29it/s]
89%|██████████████████████████████████████████████████████████████████▍
| 31836/35912 [01:50<00:14, 279.43it/s]
89%|███████████████████████████████████████████████████████████████████
| 31865/35912 [01:50<00:14, 279.48it/s]
89%|███████████████████████████████████████████████████████████████████
| 31894/35912 [01:50<00:14, 274.76it/s]
89%|███████████████████████████████████████████████████████████████████▎
| 31924/35912 [01:50<00:14, 281.30it/s]
89%|███████████████████████████████████████████████████████████████████▎
| 31956/35912 [01:50<00:13, 291.30it/s]
89%|███████████████████████████████████████████████████████████████████▍
| 31986/35912 [01:50<00:14, 279.34it/s]
89%|███████████████████████████████████████████████████████████████████▍
| 32015/35912 [01:50<00:14, 267.84it/s]
89%|███████████████████████████████████████████████████████████████████▌
| 32045/35912 [01:50<00:14, 273.91it/s]
89%|███████████████████████████████████████████████████████████████████▌
| 32074/35912 [01:50<00:13, 277.97it/s]
89%|████████████████████████████████████████████████████████████████████
| 32103/35912 [01:51<00:13, 278.45it/s]
89%|████████████████████████████████████████████████████████████████████
| 32133/35912 [01:51<00:13, 283.99it/s]
90%|████████████████████████████████████████████████████████████████████▏
| 32163/35912 [01:51<00:13, 287.18it/s]
90%|████████████████████████████████████████████████████████████████████▏
| 32192/35912 [01:51<00:13, 285.70it/s]
90%|████████████████████████████████████████████████████████████████████▎
| 32223/35912 [01:51<00:12, 291.16it/s]
90%|████████████████████████████████████████████████████████████████████▎
| 32257/35912 [01:51<00:12, 302.06it/s]
90%|████████████████████████████████████████████████████████████████████▍
| 32288/35912 [01:51<00:12, 286.13it/s]
90%|████████████████████████████████████████████████████████████████████▍
| 32317/35912 [01:51<00:12, 283.31it/s]
90%|█████████████████████████████████████████████████████████████████████
| 32346/35912 [01:51<00:12, 281.36it/s]
90%|█████████████████████████████████████████████████████████████████████
| 32376/35912 [01:51<00:12, 285.29it/s]
90%|█████████████████████████████████████████████████████████████████████▎
| 32409/35912 [01:52<00:12, 291.31it/s]
90%|█████████████████████████████████████████████████████████████████████▍
| 32441/35912 [01:52<00:11, 298.75it/s]
90%|█████████████████████████████████████████████████████████████████████▍
| 32471/35912 [01:52<00:12, 271.77it/s]
90%|█████████████████████████████████████████████████████████████████████▍
| 32499/35912 [01:52<00:13, 257.78it/s]
91%|█████████████████████████████████████████████████████████████████████▌
| 32528/35912 [01:52<00:12, 264.68it/s]
91%|█████████████████████████████████████████████████████████████████████▌
```

```
| 32559/35912 [01:52<00:12, 276.26it/s]
 91%|
| 32593/35912 [01:52<00:11, 289.95it/s]
 91%|
| 32623/35912 [01:52<00:11, 285.61it/s]
 91%|
| 32652/35912 [01:52<00:11, 278.88it/s]
 91%|
| 32681/35912 [01:53<00:11, 272.81it/s]
 91%|
| 32715/35912 [01:53<00:11, 285.10it/s]
 91%|
| 32746/35912 [01:53<00:11, 284.34it/s]
 91%|
| 32775/35912 [01:53<00:11, 284.56it/s]
 91%|
| 32807/35912 [01:53<00:10, 292.14it/s]
 91%|
| 32844/35912 [01:53<00:09, 310.47it/s]
 92%|
| 32876/35912 [01:53<00:10, 302.01it/s]
 92%|
| 32907/35912 [01:53<00:10, 284.52it/s]
 92%|
| 32937/35912 [01:53<00:10, 287.56it/s]
 92%|
| 32967/35912 [01:54<00:10, 288.89it/s]
 92%|
| 32997/35912 [01:54<00:10, 288.16it/s]
 92%|
| 33026/35912 [01:54<00:10, 277.37it/s]
 92%|
| 33059/35912 [01:54<00:09, 290.74it/s]
 92%|
| 33093/35912 [01:54<00:09, 302.55it/s]
 92%|
| 33124/35912 [01:54<00:09, 292.92it/s]
 92%|
| 33157/35912 [01:54<00:09, 302.50it/s]
 92%|
| 33188/35912 [01:54<00:09, 296.26it/s]
 93%|
| 33223/35912 [01:54<00:08, 307.52it/s]
 93%|
| 33255/35912 [01:54<00:08, 302.59it/s]
 93%|
| 33286/35912 [01:55<00:08, 292.12it/s]
 93%|
| 33316/35912 [01:55<00:08, 289.56it/s]
 93%|
| 33346/35912 [01:55<00:08, 286.98it/s]
 93%|
| 33381/35912 [01:55<00:08, 302.79it/s]
 93%|
| 33414/35912 [01:55<00:08, 303.86it/s]
 93%|
| 33445/35912 [01:55<00:08, 281.79it/s]
```

```
93%|█
| 33475/35912 [01:55<00:08, 285.60it/s]
93%|█
| 33504/35912 [01:55<00:08, 278.07it/s]
93%|█
| 33536/35912 [01:55<00:08, 287.32it/s]
93%|█
| 33566/35912 [01:56<00:08, 279.08it/s]
94%|█
| 33597/35912 [01:56<00:08, 284.74it/s]
94%|█
| 33626/35912 [01:56<00:08, 280.71it/s]
94%|█
| 33655/35912 [01:56<00:08, 281.19it/s]
94%|█
| 33684/35912 [01:56<00:08, 276.70it/s]
94%|█
| 33713/35912 [01:56<00:07, 278.36it/s]
94%|█
| 33743/35912 [01:56<00:07, 283.93it/s]
94%|█
| 33772/35912 [01:56<00:08, 264.13it/s]
94%|█
| 33802/35912 [01:56<00:07, 272.66it/s]
94%|█
| 33830/35912 [01:57<00:07, 273.43it/s]
94%|█
| 33862/35912 [01:57<00:07, 284.60it/s]
94%|█
| 33893/35912 [01:57<00:06, 288.73it/s]
94%|█
| 33927/35912 [01:57<00:06, 301.83it/s]
95%|█
| 33958/35912 [01:57<00:06, 300.93it/s]
95%|█
| 33989/35912 [01:57<00:06, 296.03it/s]
95%|█
| 34020/35912 [01:57<00:06, 298.59it/s]
95%|█
| 34050/35912 [01:57<00:06, 287.25it/s]
95%|█
| 34084/35912 [01:57<00:06, 300.68it/s]
95%|█
| 34119/35912 [01:57<00:05, 305.96it/s]
95%|█
| 34150/35912 [01:58<00:05, 302.91it/s]
95%|█
| 34181/35912 [01:58<00:05, 289.07it/s]
95%|█
| 34211/35912 [01:58<00:05, 290.76it/s]
95%|█
| 34242/35912 [01:58<00:05, 290.71it/s]
95%|█
| 34272/35912 [01:58<00:05, 291.96it/s]
96%|█
| 34302/35912 [01:58<00:05, 278.98it/s]
96%|█
```

```
| 34337/35912 [01:58<00:05, 295.77it/s]
 96%|█████████████████████████████████████████████▋
| 34367/35912 [01:58<00:05, 291.21it/s]
 96%|█████████████████████████████████████████████▋
| 34397/35912 [01:58<00:05, 293.16it/s]
 96%|█████████████████████████████████████████████▋
| 34429/35912 [01:59<00:04, 300.12it/s]
 96%|█████████████████████████████████████████████▋
| 34460/35912 [01:59<00:04, 298.88it/s]
 96%|██████████████████████████████████████████████
| 34491/35912 [01:59<00:04, 284.91it/s]
 96%|██████████████████████████████████████████████
| 34521/35912 [01:59<00:04, 287.83it/s]
 96%|██████████████████████████████████████████████
▏   | 34550/35912 [01:59<00:04, 286.15it/s]
 96%|██████████████████████████████████████████████
▏   | 34579/35912 [01:59<00:04, 281.67it/s]
 96%|██████████████████████████████████████████████
▎   | 34613/35912 [01:59<00:04, 296.36it/s]
 96%|██████████████████████████████████████████████
▎   | 34644/35912 [01:59<00:04, 298.86it/s]
 97%|██████████████████████████████████████████████
▍   | 34675/35912 [01:59<00:04, 298.86it/s]
 97%|██████████████████████████████████████████████
▍   | 34706/35912 [01:59<00:04, 301.48it/s]
 97%|██████████████████████████████████████████████
▎  | 34738/35912 [02:00<00:03, 306.16it/s]
 97%|██████████████████████████████████████████████
▎  | 34772/35912 [02:00<00:03, 311.49it/s]
 97%|██████████████████████████████████████████████
▋   | 34804/35912 [02:00<00:03, 312.41it/s]
 97%|██████████████████████████████████████████████
▊   | 34836/35912 [02:00<00:03, 296.56it/s]
 97%|██████████████████████████████████████████████
▊   | 34866/35912 [02:00<00:03, 290.91it/s]
 97%|██████████████████████████████████████████████
▉   | 34898/35912 [02:00<00:03, 296.80it/s]
 97%|██████████████████████████████████████████████
▉   | 34935/35912 [02:00<00:03, 314.93it/s]
 97%|██████████████████████████████████████████████
▌  | 34967/35912 [02:00<00:03, 313.90it/s]
 97%|██████████████████████████████████████████████
▌  | 34999/35912 [02:00<00:03, 291.82it/s]
 98%|██████████████████████████████████████████████
▊   | 35029/35912 [02:01<00:03, 291.02it/s]
 98%|██████████████████████████████████████████████
▊   | 35059/35912 [02:01<00:02, 289.65it/s]
 98%|██████████████████████████████████████████████
▉   | 35091/35912 [02:01<00:02, 297.53it/s]
 98%|██████████████████████████████████████████████
▉   | 35121/35912 [02:01<00:02, 269.61it/s]
 98%|██████████████████████████████████████████████
█   | 35149/35912 [02:01<00:02, 272.07it/s]
 98%|██████████████████████████████████████████████
█   | 35178/35912 [02:01<00:02, 275.84it/s]
 98%|██████████████████████████████████████████████
▉  | 35206/35912 [02:01<00:02, 274.85it/s]
```

```
 98%|████████████████████████████████████████████████████████████████████████████
█ | 35234/35912 [02:01<00:02, 266.36it/s]
 98%|████████████████████████████████████████████████████████████████████████████
██ | 35265/35912 [02:01<00:02, 276.82it/s]
 98%|████████████████████████████████████████████████████████████████████████████
██ | 35295/35912 [02:02<00:02, 281.22it/s]
 98%|████████████████████████████████████████████████████████████████████████████
██ | 35332/35912 [02:02<00:01, 298.83it/s]
 98%|████████████████████████████████████████████████████████████████████████████
██ | 35363/35912 [02:02<00:01, 292.93it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35393/35912 [02:02<00:01, 285.17it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35423/35912 [02:02<00:01, 288.02it/s]
 99%|████████████████████████████████████████████████████████████████████████████
█ | 35453/35912 [02:02<00:01, 290.05it/s]
 99%|████████████████████████████████████████████████████████████████████████████
█ | 35483/35912 [02:02<00:01, 285.68it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35512/35912 [02:02<00:01, 283.00it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35547/35912 [02:02<00:01, 298.91it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35579/35912 [02:02<00:01, 298.36it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35611/35912 [02:03<00:00, 302.19it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35642/35912 [02:03<00:00, 285.42it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██ | 35671/35912 [02:03<00:00, 284.48it/s]
 99%|████████████████████████████████████████████████████████████████████████████
██| 35700/35912 [02:03<00:00, 279.73it/s]
100%|████████████████████████████████████████████████████████████████████████████
███| 35733/35912 [02:03<00:00, 287.97it/s]
100%|████████████████████████████████████████████████████████████████████████████
███| 35769/35912 [02:03<00:00, 305.79it/s]
100%|████████████████████████████████████████████████████████████████████████████
███| 35803/35912 [02:03<00:00, 313.80it/s]
100%|████████████████████████████████████████████████████████████████████████████
███| 35835/35912 [02:03<00:00, 305.94it/s]
100%|████████████████████████████████████████████████████████████████████████████
████| 35866/35912 [02:03<00:00, 299.40it/s]
100%|████████████████████████████████████████████████████████████████████████████
███| 35912/35912 [02:04<00:00, 289.35it/s]
```

```
35912
300
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-116-77925bdae2d3> in <module>
     18 print(len(tfidf_w2v_vectors_train))
     19 print(len(tfidf_w2v_vectors_train[0]))
---> 20 print(tfidf_w2v_vectors_train.shape)

AttributeError: 'list' object has no attribute 'shape'
```

In [117]:
```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_cv = []; # the avg-w2v for each sentence/review is stored in
this list
for sentence in tqdm(X_cv['essay']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/revie
w
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf v
alue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_cv.append(vector)

print(len(tfidf_w2v_vectors_cv))
print(len(tfidf_w2v_vectors_cv[0]))
```

```
  0%|
| 0/17688 [00:00<?, ?it/s]
  0%|▏
| 30/17688 [00:00<00:59, 294.90it/s]
  0%|▎
| 60/17688 [00:00<00:59, 294.04it/s]
  1%|▍
| 90/17688 [00:00<00:59, 294.30it/s]
  1%|▍
| 121/17688 [00:00<00:58, 298.21it/s]
  1%|▌
| 146/17688 [00:00<01:02, 281.21it/s]
  1%|▋
| 173/17688 [00:00<01:03, 275.43it/s]
  1%|▋
| 199/17688 [00:00<01:04, 269.16it/s]
  1%|▌
| 232/17688 [00:00<01:01, 284.38it/s]
  1%|▋
| 263/17688 [00:00<01:00, 290.20it/s]
  2%|▊
| 295/17688 [00:01<00:58, 297.93it/s]
  2%|▉
| 328/17688 [00:01<00:56, 306.25it/s]
  2%|▊
| 360/17688 [00:01<00:56, 307.81it/s]
  2%|█
| 391/17688 [00:01<00:58, 294.66it/s]
  2%|█▏
| 423/17688 [00:01<00:57, 301.21it/s]
  3%|█▎
| 454/17688 [00:01<00:57, 300.51it/s]
  3%|█
| 485/17688 [00:01<01:00, 285.94it/s]
  3%|█▏
| 516/17688 [00:01<00:58, 291.33it/s]
  3%|█▍
| 546/17688 [00:01<01:03, 268.15it/s]
  3%|█▎
| 579/17688 [00:01<01:00, 282.86it/s]
  3%|█▌
| 612/17688 [00:02<00:57, 294.94it/s]
  4%|█▋
| 643/17688 [00:02<01:01, 278.60it/s]
  4%|█▊
| 675/17688 [00:02<00:59, 286.95it/s]
  4%|█▌
| 705/17688 [00:02<01:01, 278.06it/s]
  4%|█▋
| 738/17688 [00:02<00:58, 291.27it/s]
  4%|█▊
| 768/17688 [00:02<00:57, 292.35it/s]
  5%|█▉
| 798/17688 [00:02<00:57, 291.41it/s]
  5%|█▊
| 832/17688 [00:02<00:55, 302.25it/s]
  5%|██▏
```

```
| 863/17688 [00:02<00:55, 302.99it/s]
  5%|████▌
| 896/17688 [00:03<00:54, 308.24it/s]
  5%|████
| 928/17688 [00:03<00:53, 311.02it/s]
  5%|████▏
| 960/17688 [00:03<00:53, 311.17it/s]
  6%|████▍
| 994/17688 [00:03<00:52, 315.97it/s]
  6%|████▋
| 1026/17688 [00:03<00:54, 305.63it/s]
  6%|████▌
| 1057/17688 [00:03<00:58, 282.08it/s]
  6%|█████▏
| 1088/17688 [00:03<00:57, 286.92it/s]
  6%|█████▍
| 1120/17688 [00:03<00:56, 295.50it/s]
  7%|█████▌
| 1150/17688 [00:03<00:57, 285.24it/s]
  7%|█████
| 1184/17688 [00:04<00:55, 296.80it/s]
  7%|█████▏
| 1216/17688 [00:04<00:55, 298.54it/s]
  7%|█████▍
| 1247/17688 [00:04<00:56, 291.09it/s]
  7%|█████▋
| 1279/17688 [00:04<00:55, 297.76it/s]
  7%|██████▏
| 1314/17688 [00:04<00:52, 310.28it/s]
  8%|██████▍
| 1346/17688 [00:04<00:52, 308.86it/s]
  8%|██████▌
| 1378/17688 [00:04<00:58, 280.42it/s]
  8%|██████
| 1407/17688 [00:04<00:58, 277.76it/s]
  8%|██████▏
| 1440/17688 [00:04<00:55, 291.05it/s]
  8%|██████▍
| 1470/17688 [00:05<00:55, 292.19it/s]
  8%|██████▋
| 1502/17688 [00:05<00:54, 298.55it/s]
  9%|██████▌
| 1533/17688 [00:05<00:54, 297.80it/s]
  9%|███████▏
| 1563/17688 [00:05<00:54, 296.90it/s]
  9%|███████▍
| 1593/17688 [00:05<00:56, 283.75it/s]
  9%|███████▋
| 1625/17688 [00:05<00:54, 293.14it/s]
  9%|███████
| 1655/17688 [00:05<00:54, 291.96it/s]
 10%|███████▏
| 1685/17688 [00:05<00:55, 287.79it/s]
 10%|███████▍
| 1715/17688 [00:05<00:55, 289.89it/s]
 10%|███████▋
| 1745/17688 [00:05<00:54, 290.53it/s]
```

```
 10%|████████
| 1775/17688 [00:06<00:57, 278.07it/s]
 10%|█████████
| 1805/17688 [00:06<00:55, 283.72it/s]
 10%|█████████
| 1834/17688 [00:06<00:55, 284.12it/s]
 11%|████████
| 1871/17688 [00:06<00:51, 304.84it/s]
 11%|████████
| 1904/17688 [00:06<00:52, 302.78it/s]
 11%|█████████
| 1935/17688 [00:06<00:52, 300.72it/s]
 11%|█████████
| 1973/17688 [00:06<00:49, 318.60it/s]
 11%|█████████
| 2006/17688 [00:06<00:50, 310.40it/s]
 12%|██████████
| 2039/17688 [00:06<00:49, 315.37it/s]
 12%|██████████
| 2071/17688 [00:07<00:50, 306.99it/s]
 12%|█████████
| 2103/17688 [00:07<00:50, 309.22it/s]
 12%|█████████
| 2135/17688 [00:07<00:51, 301.18it/s]
 12%|██████████
| 2166/17688 [00:07<00:53, 292.84it/s]
 12%|██████████
| 2196/17688 [00:07<00:54, 286.74it/s]
 13%|██████████
| 2225/17688 [00:07<00:55, 276.44it/s]
 13%|██████████
| 2257/17688 [00:07<00:53, 286.11it/s]
 13%|███████████
| 2288/17688 [00:07<00:52, 291.45it/s]
 13%|███████████
| 2318/17688 [00:07<00:55, 277.12it/s]
 13%|██████████
| 2346/17688 [00:07<00:57, 267.08it/s]
 13%|██████████
| 2376/17688 [00:08<00:56, 268.96it/s]
 14%|███████████
| 2404/17688 [00:08<00:56, 270.82it/s]
 14%|███████████
| 2435/17688 [00:08<00:54, 279.42it/s]
 14%|███████████
| 2467/17688 [00:08<00:52, 289.89it/s]
 14%|███████████
| 2500/17688 [00:08<00:50, 298.64it/s]
 14%|████████████
| 2533/17688 [00:08<00:49, 305.92it/s]
 14%|███████████
| 2564/17688 [00:08<00:52, 289.35it/s]
 15%|███████████
| 2594/17688 [00:08<00:52, 287.66it/s]
 15%|████████████
| 2623/17688 [00:08<00:53, 283.52it/s]
 15%|████████████
```

```
| 2652/17688 [00:09<00:53, 279.05it/s]
 15%|
| 2681/17688 [00:09<00:54, 274.49it/s]
 15%|
| 2711/17688 [00:09<00:53, 279.53it/s]
 16%|
| 2743/17688 [00:09<00:51, 287.64it/s]
 16%|
| 2776/17688 [00:09<00:50, 293.80it/s]
 16%|
| 2806/17688 [00:09<00:53, 279.38it/s]
 16%|
| 2835/17688 [00:09<00:53, 275.47it/s]
 16%|
| 2863/17688 [00:09<00:54, 272.20it/s]
 16%|
| 2891/17688 [00:09<00:55, 267.64it/s]
 17%|
| 2921/17688 [00:10<00:53, 275.27it/s]
 17%|
| 2954/17688 [00:10<00:51, 287.62it/s]
 17%|
| 2987/17688 [00:10<00:49, 296.94it/s]
 17%|
| 3018/17688 [00:10<00:48, 300.10it/s]
 17%|
| 3050/17688 [00:10<00:48, 303.44it/s]
 17%|
| 3085/17688 [00:10<00:46, 313.74it/s]
 18%|
| 3119/17688 [00:10<00:45, 317.83it/s]
 18%|
| 3151/17688 [00:10<00:46, 312.22it/s]
 18%|
| 3183/17688 [00:10<00:47, 305.76it/s]
 18%|
| 3214/17688 [00:10<00:47, 303.66it/s]
 18%|
| 3246/17688 [00:11<00:47, 305.10it/s]
 19%|
| 3279/17688 [00:11<00:46, 310.64it/s]
 19%|
| 3312/17688 [00:11<00:45, 315.55it/s]
 19%|
| 3344/17688 [00:11<00:47, 299.37it/s]
 19%|
| 3375/17688 [00:11<00:48, 297.50it/s]
 19%|
| 3405/17688 [00:11<00:48, 295.84it/s]
 19%|
| 3435/17688 [00:11<00:49, 290.39it/s]
 20%|
| 3465/17688 [00:11<00:49, 287.58it/s]
 20%|
| 3495/17688 [00:11<00:49, 288.90it/s]
 20%|
| 3524/17688 [00:12<00:49, 286.05it/s]
```

```
 20%|
| 3553/17688 [00:12<00:50, 279.17it/s]
 20%|
| 3581/17688 [00:12<00:53, 261.66it/s]
 20%|
| 3611/17688 [00:12<00:51, 270.80it/s]
 21%|
| 3644/17688 [00:12<00:49, 285.68it/s]
 21%|
| 3673/17688 [00:12<00:49, 280.54it/s]
 21%|
| 3704/17688 [00:12<00:48, 288.18it/s]
 21%|
| 3734/17688 [00:12<00:48, 287.65it/s]
 21%|
| 3763/17688 [00:12<00:48, 286.89it/s]
 21%|
| 3792/17688 [00:12<00:48, 285.50it/s]
 22%|
| 3821/17688 [00:13<00:49, 279.61it/s]
 22%|
| 3851/17688 [00:13<00:48, 283.23it/s]
 22%|
| 3883/17688 [00:13<00:47, 291.95it/s]
 22%|
| 3913/17688 [00:13<00:48, 285.33it/s]
 22%|
| 3942/17688 [00:13<00:48, 286.10it/s]
 22%|
| 3971/17688 [00:13<00:49, 276.81it/s]
 23%|
| 4000/17688 [00:13<00:49, 279.22it/s]
 23%|
| 4029/17688 [00:13<00:50, 269.26it/s]
 23%|
| 4058/17688 [00:13<00:49, 274.59it/s]
 23%|
| 4094/17688 [00:14<00:46, 290.82it/s]
 23%|
| 4124/17688 [00:14<00:46, 292.89it/s]
 23%|
| 4154/17688 [00:14<00:48, 278.03it/s]
 24%|
| 4184/17688 [00:14<00:47, 282.09it/s]
 24%|
| 4218/17688 [00:14<00:45, 294.40it/s]
 24%|
| 4248/17688 [00:14<00:46, 291.13it/s]
 24%|
| 4280/17688 [00:14<00:45, 297.79it/s]
 24%|
| 4314/17688 [00:14<00:44, 302.13it/s]
 25%|
| 4345/17688 [00:14<00:43, 303.79it/s]
 25%|
| 4376/17688 [00:14<00:44, 302.30it/s]
 25%|
```

```
| 4407/17688 [00:15<00:44, 296.10it/s]
25%|
| 4437/17688 [00:15<00:46, 284.82it/s]
25%|
| 4472/17688 [00:15<00:44, 295.03it/s]
25%|
| 4502/17688 [00:15<00:45, 291.56it/s]
26%|
| 4532/17688 [00:15<00:44, 293.41it/s]
26%|
| 4562/17688 [00:15<00:44, 292.15it/s]
26%|
| 4592/17688 [00:15<00:45, 285.46it/s]
26%|
| 4627/17688 [00:15<00:43, 299.29it/s]
26%|
| 4658/17688 [00:15<00:45, 285.96it/s]
26%|
| 4687/17688 [00:16<00:47, 275.93it/s]
27%|
| 4719/17688 [00:16<00:45, 285.72it/s]
27%|
| 4748/17688 [00:16<00:47, 274.22it/s]
27%|
| 4779/17688 [00:16<00:46, 280.42it/s]
27%|
| 4813/17688 [00:16<00:43, 294.65it/s]
27%|
| 4846/17688 [00:16<00:42, 302.15it/s]
28%|
| 4877/17688 [00:16<00:44, 286.98it/s]
28%|
| 4910/17688 [00:16<00:43, 293.31it/s]
28%|
| 4940/17688 [00:16<00:43, 292.08it/s]
28%|
| 4970/17688 [00:17<00:45, 282.20it/s]
28%|
| 4999/17688 [00:17<00:45, 276.60it/s]
28%|
| 5030/17688 [00:17<00:44, 281.38it/s]
29%|
| 5060/17688 [00:17<00:44, 285.31it/s]
29%|
| 5093/17688 [00:17<00:42, 296.01it/s]
29%|
| 5123/17688 [00:17<00:42, 293.94it/s]
29%|
| 5153/17688 [00:17<00:43, 290.82it/s]
29%|
| 5183/17688 [00:17<00:42, 292.03it/s]
29%|
| 5213/17688 [00:17<00:42, 292.03it/s]
30%|
| 5245/17688 [00:17<00:41, 298.45it/s]
30%|
| 5275/17688 [00:18<00:42, 290.48it/s]
```

```
 30%|███████████████████████
| 5305/17688 [00:18<00:42, 291.80it/s]
 30%|███████████████████████
| 5335/17688 [00:18<00:44, 275.06it/s]
 30%|███████████████████████
| 5365/17688 [00:18<00:43, 280.72it/s]
 31%|███████████████████████
| 5395/17688 [00:18<00:43, 284.03it/s]
 31%|███████████████████████
| 5424/17688 [00:18<00:44, 277.82it/s]
 31%|███████████████████████
| 5454/17688 [00:18<00:43, 283.53it/s]
 31%|███████████████████████
| 5486/17688 [00:18<00:41, 291.38it/s]
 31%|████████████████████████
| 5516/17688 [00:18<00:43, 279.39it/s]
 31%|████████████████████████
| 5545/17688 [00:19<00:43, 281.07it/s]
 32%|████████████████████████
| 5579/17688 [00:19<00:40, 295.92it/s]
 32%|████████████████████████
| 5609/17688 [00:19<00:40, 294.75it/s]
 32%|████████████████████████
| 5643/17688 [00:19<00:39, 306.38it/s]
 32%|████████████████████████
| 5674/17688 [00:19<00:40, 294.60it/s]
 32%|████████████████████████
| 5704/17688 [00:19<00:40, 295.56it/s]
 32%|█████████████████████████
| 5734/17688 [00:19<00:40, 295.36it/s]
 33%|█████████████████████████
| 5764/17688 [00:19<00:41, 286.78it/s]
 33%|█████████████████████████
| 5794/17688 [00:19<00:41, 290.01it/s]
 33%|█████████████████████████
| 5825/17688 [00:19<00:40, 295.10it/s]
 33%|█████████████████████████
| 5855/17688 [00:20<00:40, 289.10it/s]
 33%|█████████████████████████
| 5891/17688 [00:20<00:38, 305.90it/s]
 33%|█████████████████████████
| 5924/17688 [00:20<00:38, 308.61it/s]
 34%|█████████████████████████
| 5958/17688 [00:20<00:37, 315.00it/s]
 34%|██████████████████████████
| 5990/17688 [00:20<00:38, 301.55it/s]
 34%|██████████████████████████
| 6021/17688 [00:20<00:39, 298.99it/s]
 34%|██████████████████████████
| 6055/17688 [00:20<00:37, 307.94it/s]
 34%|██████████████████████████
| 6086/17688 [00:20<00:38, 303.37it/s]
 35%|██████████████████████████
| 6117/17688 [00:20<00:37, 304.67it/s]
 35%|██████████████████████████
| 6148/17688 [00:21<00:38, 296.84it/s]
 35%|██████████████████████████
```

```
| 6178/17688 [00:21<00:39, 288.58it/s]
 35%|██████████████████████████▏
| 6207/17688 [00:21<00:39, 287.52it/s]
 35%|██████████████████████████▍
| 6239/17688 [00:21<00:38, 294.32it/s]
 35%|██████████████████████████▌
| 6270/17688 [00:21<00:38, 297.37it/s]
 36%|███████████████████████████
| 6301/17688 [00:21<00:37, 300.41it/s]
 36%|███████████████████████████▏
| 6335/17688 [00:21<00:36, 307.29it/s]
 36%|███████████████████████████▍
| 6366/17688 [00:21<00:37, 301.18it/s]
 36%|███████████████████████████▌
| 6397/17688 [00:21<00:39, 286.37it/s]
 36%|███████████████████████████▋
| 6430/17688 [00:21<00:38, 296.01it/s]
 37%|███████████████████████████▉
| 6460/17688 [00:22<00:38, 291.38it/s]
 37%|████████████████████████████
| 6490/17688 [00:22<00:39, 283.34it/s]
 37%|████████████████████████████
| 6519/17688 [00:22<00:40, 277.36it/s]
 37%|████████████████████████████▏
| 6547/17688 [00:22<00:40, 277.54it/s]
 37%|████████████████████████████▍
| 6579/17688 [00:22<00:38, 288.47it/s]
 37%|████████████████████████████▌
| 6609/17688 [00:22<00:38, 289.54it/s]
 38%|████████████████████████████▋
| 6641/17688 [00:22<00:37, 297.44it/s]
 38%|████████████████████████████▉
| 6671/17688 [00:22<00:38, 289.82it/s]
 38%|█████████████████████████████
| 6705/17688 [00:22<00:36, 301.05it/s]
 38%|█████████████████████████████▏
| 6737/17688 [00:23<00:36, 304.11it/s]
 38%|█████████████████████████████
| 6768/17688 [00:23<00:35, 303.41it/s]
 38%|█████████████████████████████▏
| 6799/17688 [00:23<00:35, 302.92it/s]
 39%|█████████████████████████████▍
| 6830/17688 [00:23<00:36, 294.82it/s]
 39%|█████████████████████████████▌
| 6860/17688 [00:23<00:37, 291.42it/s]
 39%|█████████████████████████████▋
| 6890/17688 [00:23<00:38, 278.62it/s]
 39%|█████████████████████████████▉
| 6923/17688 [00:23<00:37, 287.92it/s]
 39%|██████████████████████████████
| 6952/17688 [00:23<00:37, 286.21it/s]
 39%|██████████████████████████████▏
| 6983/17688 [00:23<00:37, 289.09it/s]
 40%|██████████████████████████████▎
| 7013/17688 [00:24<00:36, 290.81it/s]
 40%|██████████████████████████████▍
| 7043/17688 [00:24<00:36, 288.66it/s]
```

```
40%|
| 7072/17688 [00:24<00:37, 285.85it/s]
40%|
| 7103/17688 [00:24<00:36, 292.11it/s]
40%|
| 7133/17688 [00:24<00:36, 287.90it/s]
41%|
| 7168/17688 [00:24<00:34, 303.51it/s]
41%|
| 7199/17688 [00:24<00:35, 298.61it/s]
41%|
| 7230/17688 [00:24<00:36, 288.71it/s]
41%|
| 7260/17688 [00:24<00:36, 286.39it/s]
41%|
| 7289/17688 [00:24<00:37, 277.80it/s]
41%|
| 7322/17688 [00:25<00:35, 290.31it/s]
42%|
| 7352/17688 [00:25<00:35, 290.83it/s]
42%|
| 7382/17688 [00:25<00:35, 292.04it/s]
42%|
| 7412/17688 [00:25<00:34, 293.75it/s]
42%|
| 7442/17688 [00:25<00:35, 290.68it/s]
42%|
| 7472/17688 [00:25<00:35, 288.57it/s]
42%|
| 7502/17688 [00:25<00:35, 289.61it/s]
43%|
| 7535/17688 [00:25<00:34, 298.43it/s]
43%|
| 7565/17688 [00:25<00:35, 285.51it/s]
43%|
| 7594/17688 [00:26<00:35, 281.24it/s]
43%|
| 7625/17688 [00:26<00:34, 287.87it/s]
43%|
| 7655/17688 [00:26<00:34, 289.14it/s]
43%|
| 7685/17688 [00:26<00:34, 289.99it/s]
44%|
| 7715/17688 [00:26<00:35, 279.28it/s]
44%|
| 7744/17688 [00:26<00:37, 267.06it/s]
44%|
| 7773/17688 [00:26<00:36, 272.99it/s]
44%|
| 7801/17688 [00:26<00:36, 272.85it/s]
44%|
| 7830/17688 [00:26<00:35, 274.85it/s]
44%|
| 7864/17688 [00:26<00:34, 288.85it/s]
45%|
| 7894/17688 [00:27<00:34, 285.64it/s]
45%|
```

```
| 7928/17688 [00:27<00:32, 299.48it/s]
45%|███████████████████████████████████
| 7959/17688 [00:27<00:34, 278.38it/s]
45%|███████████████████████████████████
| 7990/17688 [00:27<00:33, 286.60it/s]
45%|███████████████████████████████████
| 8022/17688 [00:27<00:32, 293.64it/s]
46%|███████████████████████████████████
| 8056/17688 [00:27<00:31, 305.56it/s]
46%|████████████████████████████████████
| 8088/17688 [00:27<00:31, 309.10it/s]
46%|████████████████████████████████████
| 8120/17688 [00:27<00:32, 292.05it/s]
46%|███████████████████████████████████
| 8150/17688 [00:27<00:32, 292.88it/s]
46%|████████████████████████████████████
| 8180/17688 [00:28<00:33, 287.61it/s]
46%|████████████████████████████████████
| 8214/17688 [00:28<00:31, 300.97it/s]
47%|████████████████████████████████████
| 8245/17688 [00:28<00:31, 295.20it/s]
47%|████████████████████████████████████
| 8276/17688 [00:28<00:31, 298.86it/s]
47%|█████████████████████████████████████
| 8307/17688 [00:28<00:31, 298.86it/s]
47%|█████████████████████████████████████
| 8338/17688 [00:28<00:31, 300.57it/s]
47%|█████████████████████████████████████
| 8369/17688 [00:28<00:31, 292.46it/s]
48%|████████████████████████████████████
| 8403/17688 [00:28<00:30, 303.83it/s]
48%|█████████████████████████████████████
| 8434/17688 [00:28<00:30, 301.47it/s]
48%|█████████████████████████████████████
| 8465/17688 [00:28<00:32, 288.14it/s]
48%|█████████████████████████████████████
| 8496/17688 [00:29<00:31, 292.10it/s]
48%|██████████████████████████████████████
| 8526/17688 [00:29<00:31, 290.39it/s]
48%|██████████████████████████████████████
| 8556/17688 [00:29<00:31, 292.58it/s]
49%|██████████████████████████████████████
| 8586/17688 [00:29<00:31, 289.05it/s]
49%|█████████████████████████████████████
| 8618/17688 [00:29<00:30, 297.08it/s]
49%|██████████████████████████████████████
| 8648/17688 [00:29<00:31, 286.26it/s]
49%|██████████████████████████████████████
| 8679/17688 [00:29<00:30, 291.57it/s]
49%|██████████████████████████████████████
| 8711/17688 [00:29<00:30, 295.64it/s]
49%|██████████████████████████████████████
| 8741/17688 [00:29<00:30, 296.29it/s]
50%|███████████████████████████████████████
| 8771/17688 [00:30<00:30, 291.57it/s]
50%|███████████████████████████████████████
| 8808/17688 [00:30<00:28, 308.47it/s]
```

```
 50%|█████████████████████████████▌
| 8840/17688 [00:30<00:30, 290.87it/s]
 50%|██████████████████████████████
| 8870/17688 [00:30<00:30, 292.05it/s]
 50%|██████████████████████████████▏
| 8900/17688 [00:30<00:30, 284.60it/s]
 50%|██████████████████████████████▎
| 8929/17688 [00:30<00:31, 278.21it/s]
 51%|██████████████████████████████
| 8961/17688 [00:30<00:30, 287.42it/s]
 51%|██████████████████████████████▌
| 8990/17688 [00:30<00:32, 269.20it/s]
 51%|██████████████████████████████▋
| 9019/17688 [00:30<00:31, 273.77it/s]
 51%|██████████████████████████████▊
| 9050/17688 [00:31<00:30, 281.59it/s]
 51%|███████████████████████████████
| 9079/17688 [00:31<00:31, 270.05it/s]
 51%|███████████████████████████████▏
| 9109/17688 [00:31<00:30, 277.05it/s]
 52%|███████████████████████████████▎
| 9139/17688 [00:31<00:30, 282.17it/s]
 52%|███████████████████████████████▍
| 9168/17688 [00:31<00:32, 264.50it/s]
 52%|███████████████████████████████
| 9198/17688 [00:31<00:31, 273.69it/s]
 52%|███████████████████████████████▌
| 9226/17688 [00:31<00:31, 271.77it/s]
 52%|███████████████████████████████▋
| 9257/17688 [00:31<00:30, 279.37it/s]
 52%|███████████████████████████████▊
| 9286/17688 [00:31<00:30, 277.83it/s]
 53%|████████████████████████████████
| 9314/17688 [00:31<00:30, 277.87it/s]
 53%|████████████████████████████████▏
| 9344/17688 [00:32<00:29, 280.40it/s]
 53%|████████████████████████████████▎
| 9374/17688 [00:32<00:29, 282.99it/s]
 53%|████████████████████████████████▍
| 9403/17688 [00:32<00:29, 277.92it/s]
 53%|████████████████████████████████
| 9431/17688 [00:32<00:30, 273.87it/s]
 53%|████████████████████████████████▌
| 9462/17688 [00:32<00:29, 279.40it/s]
 54%|████████████████████████████████▋
| 9496/17688 [00:32<00:28, 290.86it/s]
 54%|████████████████████████████████▊
| 9526/17688 [00:32<00:28, 288.69it/s]
 54%|█████████████████████████████████
| 9556/17688 [00:32<00:27, 290.53it/s]
 54%|█████████████████████████████████▏
| 9592/17688 [00:32<00:26, 306.24it/s]
 54%|█████████████████████████████████▎
| 9623/17688 [00:33<00:27, 288.74it/s]
 55%|█████████████████████████████████▍
| 9653/17688 [00:33<00:28, 286.41it/s]
 55%|█████████████████████████████████
```

```
| 9686/17688 [00:33<00:27, 296.04it/s]
55%|██████████████████████████████████████████████████████
| 9721/17688 [00:33<00:25, 307.35it/s]
55%|██████████████████████████████████████████████████████
| 9753/17688 [00:33<00:26, 302.48it/s]
55%|██████████████████████████████████████████████████████
| 9784/17688 [00:33<00:27, 292.04it/s]
55%|██████████████████████████████████████████████████████
| 9814/17688 [00:33<00:28, 276.73it/s]
56%|██████████████████████████████████████████████████████
| 9848/17688 [00:33<00:26, 291.79it/s]
56%|██████████████████████████████████████████████████████
| 9879/17688 [00:33<00:26, 293.88it/s]
56%|██████████████████████████████████████████████████████
| 9909/17688 [00:34<00:28, 275.57it/s]
56%|██████████████████████████████████████████████████████
| 9940/17688 [00:34<00:27, 284.52it/s]
56%|██████████████████████████████████████████████████████
| 9972/17688 [00:34<00:26, 293.71it/s]
57%|██████████████████████████████████████████████████████
| 10002/17688 [00:34<00:26, 289.82it/s]
57%|██████████████████████████████████████████████████████
| 10032/17688 [00:34<00:27, 273.82it/s]
57%|██████████████████████████████████████████████████████
| 10065/17688 [00:34<00:26, 287.25it/s]
57%|██████████████████████████████████████████████████████
| 10095/17688 [00:34<00:26, 288.68it/s]
57%|██████████████████████████████████████████████████████
| 10125/17688 [00:34<00:26, 288.84it/s]
57%|██████████████████████████████████████████████████████
| 10155/17688 [00:34<00:25, 289.80it/s]
58%|██████████████████████████████████████████████████████
| 10185/17688 [00:34<00:25, 289.60it/s]
58%|██████████████████████████████████████████████████████
| 10215/17688 [00:35<00:26, 285.40it/s]
58%|██████████████████████████████████████████████████████
| 10251/17688 [00:35<00:24, 303.75it/s]
58%|██████████████████████████████████████████████████████
| 10282/17688 [00:35<00:25, 292.01it/s]
58%|██████████████████████████████████████████████████████
| 10312/17688 [00:35<00:25, 286.22it/s]
58%|██████████████████████████████████████████████████████
| 10345/17688 [00:35<00:24, 295.90it/s]
59%|██████████████████████████████████████████████████████
| 10375/17688 [00:35<00:25, 281.49it/s]
59%|██████████████████████████████████████████████████████
| 10404/17688 [00:35<00:26, 277.70it/s]
59%|██████████████████████████████████████████████████████
| 10433/17688 [00:35<00:26, 271.26it/s]
59%|██████████████████████████████████████████████████████
| 10461/17688 [00:35<00:27, 261.77it/s]
59%|██████████████████████████████████████████████████████
| 10495/17688 [00:36<00:25, 279.98it/s]
59%|██████████████████████████████████████████████████████
| 10524/17688 [00:36<00:25, 280.67it/s]
60%|██████████████████████████████████████████████████████
| 10555/17688 [00:36<00:24, 287.48it/s]
```

```
60%|
| 10585/17688 [00:36<00:25, 282.33it/s]
60%|
| 10615/17688 [00:36<00:24, 285.17it/s]
60%|
| 10647/17688 [00:36<00:23, 293.40it/s]
60%|
| 10684/17688 [00:36<00:22, 312.25it/s]
61%|
| 10716/17688 [00:36<00:22, 308.43it/s]
61%|
| 10748/17688 [00:36<00:24, 288.49it/s]
61%|
| 10778/17688 [00:37<00:24, 287.06it/s]
61%|
| 10809/17688 [00:37<00:23, 291.32it/s]
61%|
| 10839/17688 [00:37<00:24, 284.91it/s]
61%|
| 10868/17688 [00:37<00:24, 272.92it/s]
62%|
| 10896/17688 [00:37<00:24, 272.02it/s]
62%|
| 10927/17688 [00:37<00:24, 281.08it/s]
62%|
| 10956/17688 [00:37<00:23, 281.44it/s]
62%|
| 10988/17688 [00:37<00:23, 289.05it/s]
62%|
| 11024/17688 [00:37<00:21, 306.64it/s]
63%|
| 11056/17688 [00:37<00:22, 297.79it/s]
63%|
| 11087/17688 [00:38<00:22, 298.12it/s]
63%|
| 11118/17688 [00:38<00:22, 296.64it/s]
63%|
| 11148/17688 [00:38<00:22, 296.99it/s]
63%|
| 11178/17688 [00:38<00:23, 280.60it/s]
63%|
| 11208/17688 [00:38<00:22, 284.74it/s]
64%|
| 11239/17688 [00:38<00:22, 290.46it/s]
64%|
| 11269/17688 [00:38<00:22, 280.36it/s]
64%|
| 11303/17688 [00:38<00:21, 293.08it/s]
64%|
| 11333/17688 [00:38<00:21, 293.63it/s]
64%|
| 11363/17688 [00:39<00:21, 291.45it/s]
64%|
| 11393/17688 [00:39<00:21, 292.45it/s]
65%|
| 11423/17688 [00:39<00:21, 289.82it/s]
65%|
```

```
| 11453/17688 [00:39<00:21, 284.71it/s]
 65%|████████████████████████████████████████████████▌
| 11482/17688 [00:39<00:22, 277.48it/s]
 65%|████████████████████████████████████████████████▊
| 11515/17688 [00:39<00:21, 290.82it/s]
 65%|████████████████████████████████████████████████▉
| 11548/17688 [00:39<00:20, 300.95it/s]
 65%|█████████████████████████████████████████████████
| 11579/17688 [00:39<00:20, 292.71it/s]
 66%|█████████████████████████████████████████████████▏
| 11609/17688 [00:39<00:21, 281.82it/s]
 66%|█████████████████████████████████████████████████▎
| 11638/17688 [00:40<00:21, 278.72it/s]
 66%|█████████████████████████████████████████████████▍
| 11667/17688 [00:40<00:21, 278.16it/s]
 66%|█████████████████████████████████████████████████▌
| 11695/17688 [00:40<00:21, 274.05it/s]
 66%|█████████████████████████████████████████████████▋
| 11726/17688 [00:40<00:21, 278.79it/s]
 66%|█████████████████████████████████████████████████▊
| 11754/17688 [00:40<00:21, 276.08it/s]
 67%|█████████████████████████████████████████████████▉
| 11783/17688 [00:40<00:21, 279.52it/s]
 67%|██████████████████████████████████████████████████
| 11814/17688 [00:40<00:20, 282.72it/s]
 67%|██████████████████████████████████████████████████▏
| 11845/17688 [00:40<00:20, 288.99it/s]
 67%|██████████████████████████████████████████████████▎
| 11874/17688 [00:40<00:20, 281.94it/s]
 67%|██████████████████████████████████████████████████▍
| 11905/17688 [00:40<00:20, 288.41it/s]
 67%|██████████████████████████████████████████████████▌
| 11934/17688 [00:41<00:20, 284.87it/s]
 68%|██████████████████████████████████████████████████▋
| 11968/17688 [00:41<00:19, 298.07it/s]
 68%|██████████████████████████████████████████████████▊
| 12002/17688 [00:41<00:18, 307.24it/s]
 68%|███████████████████████████████████████████████████
| 12037/17688 [00:41<00:17, 317.43it/s]
 68%|███████████████████████████████████████████████████▏
| 12069/17688 [00:41<00:17, 317.50it/s]
 68%|███████████████████████████████████████████████████▎
| 12101/17688 [00:41<00:17, 312.91it/s]
 69%|███████████████████████████████████████████████████▍
| 12133/17688 [00:41<00:18, 307.10it/s]
 69%|███████████████████████████████████████████████████▌
| 12165/17688 [00:41<00:17, 310.18it/s]
 69%|███████████████████████████████████████████████████▋
| 12197/17688 [00:41<00:17, 311.49it/s]
 69%|███████████████████████████████████████████████████▊
| 12229/17688 [00:42<00:18, 294.37it/s]
 69%|███████████████████████████████████████████████████▉
| 12260/17688 [00:42<00:18, 298.26it/s]
 69%|████████████████████████████████████████████████████
| 12291/17688 [00:42<00:18, 293.37it/s]
 70%|████████████████████████████████████████████████████▏
| 12321/17688 [00:42<00:18, 290.42it/s]
```

```
 70%|
| 12352/17688 [00:42<00:18, 292.91it/s]
 70%|
| 12384/17688 [00:42<00:17, 298.25it/s]
 70%|
| 12414/17688 [00:42<00:18, 292.87it/s]
 70%|
| 12450/17688 [00:42<00:17, 308.08it/s]
 71%|
| 12482/17688 [00:42<00:16, 307.32it/s]
 71%|
| 12513/17688 [00:42<00:17, 290.23it/s]
 71%|
| 12543/17688 [00:43<00:17, 289.09it/s]
 71%|
| 12575/17688 [00:43<00:17, 296.29it/s]
 71%|
| 12605/17688 [00:43<00:17, 290.73it/s]
 71%|
| 12635/17688 [00:43<00:17, 289.44it/s]
 72%|
| 12669/17688 [00:43<00:16, 301.56it/s]
 72%|
| 12700/17688 [00:43<00:16, 294.76it/s]
 72%|
| 12730/17688 [00:43<00:18, 273.87it/s]
 72%|
| 12758/17688 [00:43<00:18, 265.71it/s]
 72%|
| 12786/17688 [00:43<00:18, 268.50it/s]
 72%|
| 12818/17688 [00:44<00:17, 279.37it/s]
 73%|
| 12847/17688 [00:44<00:17, 276.22it/s]
 73%|
| 12875/17688 [00:44<00:18, 261.31it/s]
 73%|
| 12904/17688 [00:44<00:17, 268.02it/s]
 73%|
| 12932/17688 [00:44<00:18, 261.83it/s]
 73%|
| 12961/17688 [00:44<00:17, 269.14it/s]
 73%|
| 12996/17688 [00:44<00:16, 288.66it/s]
 74%|
| 13026/17688 [00:44<00:16, 290.51it/s]
 74%|
| 13056/17688 [00:44<00:15, 292.66it/s]
 74%|
| 13088/17688 [00:44<00:15, 297.25it/s]
 74%|
| 13118/17688 [00:45<00:17, 268.00it/s]
 74%|
| 13152/17688 [00:45<00:15, 284.23it/s]
 75%|
| 13182/17688 [00:45<00:15, 281.70it/s]
 75%|
```

```
| 13211/17688 [00:45<00:15, 282.70it/s]
75%|██████████████████████████████████████████████████████▌
| 13243/17688 [00:45<00:15, 289.98it/s]
75%|██████████████████████████████████████████████████████▌
| 13280/17688 [00:45<00:14, 304.95it/s]
75%|██████████████████████████████████████████████████████▌
| 13311/17688 [00:45<00:15, 281.68it/s]
75%|███████████████████████████████████████████████████████
| 13342/17688 [00:45<00:15, 286.63it/s]
76%|███████████████████████████████████████████████████████
| 13372/17688 [00:46<00:15, 280.17it/s]
76%|███████████████████████████████████████████████████████
| 13401/17688 [00:46<00:15, 278.39it/s]
76%|███████████████████████████████████████████████████████
| 13430/17688 [00:46<00:15, 276.36it/s]
76%|███████████████████████████████████████████████████████▌
| 13458/17688 [00:46<00:15, 276.03it/s]
76%|███████████████████████████████████████████████████████▌
| 13486/17688 [00:46<00:15, 276.60it/s]
76%|███████████████████████████████████████████████████████▌
| 13516/17688 [00:46<00:14, 282.65it/s]
77%|███████████████████████████████████████████████████████▌
| 13546/17688 [00:46<00:14, 286.22it/s]
77%|████████████████████████████████████████████████████████
| 13579/17688 [00:46<00:14, 292.76it/s]
77%|████████████████████████████████████████████████████████
| 13609/17688 [00:46<00:14, 288.33it/s]
77%|████████████████████████████████████████████████████████
| 13638/17688 [00:46<00:14, 284.82it/s]
77%|████████████████████████████████████████████████████████
| 13667/17688 [00:47<00:14, 284.06it/s]
77%|████████████████████████████████████████████████████████▌
| 13696/17688 [00:47<00:14, 275.47it/s]
78%|████████████████████████████████████████████████████████▌
| 13726/17688 [00:47<00:14, 281.03it/s]
78%|████████████████████████████████████████████████████████▌
| 13756/17688 [00:47<00:13, 285.86it/s]
78%|████████████████████████████████████████████████████████▌
| 13785/17688 [00:47<00:13, 286.47it/s]
78%|█████████████████████████████████████████████████████████
| 13820/17688 [00:47<00:12, 301.61it/s]
78%|█████████████████████████████████████████████████████████
| 13851/17688 [00:47<00:12, 299.05it/s]
78%|█████████████████████████████████████████████████████████
| 13882/17688 [00:47<00:13, 288.19it/s]
79%|█████████████████████████████████████████████████████████
| 13914/17688 [00:47<00:12, 295.62it/s]
79%|█████████████████████████████████████████████████████████▌
| 13944/17688 [00:48<00:13, 282.90it/s]
79%|█████████████████████████████████████████████████████████▌
| 13973/17688 [00:48<00:13, 281.90it/s]
79%|█████████████████████████████████████████████████████████▌
| 14006/17688 [00:48<00:12, 294.22it/s]
79%|██████████████████████████████████████████████████████████
| 14037/17688 [00:48<00:12, 296.44it/s]
80%|██████████████████████████████████████████████████████████
| 14067/17688 [00:48<00:12, 288.31it/s]
```

```
 80%|████████████████████████████████████████▌         | 14097/17688 [00:48<00:12, 284.50it/s]
 80%|████████████████████████████████████████▌         | 14126/17688 [00:48<00:12, 285.51it/s]
 80%|████████████████████████████████████████          | 14155/17688 [00:48<00:12, 279.62it/s]
 80%|████████████████████████████████████████▏         | 14187/17688 [00:48<00:12, 287.70it/s]
 80%|████████████████████████████████████████▏         | 14216/17688 [00:48<00:12, 284.38it/s]
 81%|████████████████████████████████████████▎         | 14252/17688 [00:49<00:11, 302.19it/s]
 81%|████████████████████████████████████████▍         | 14283/17688 [00:49<00:12, 277.03it/s]
 81%|████████████████████████████████████████▌         | 14312/17688 [00:49<00:12, 278.59it/s]
 81%|████████████████████████████████████████▌         | 14346/17688 [00:49<00:11, 290.24it/s]
 81%|████████████████████████████████████████▋         | 14376/17688 [00:49<00:11, 288.27it/s]
 81%|████████████████████████████████████████▊         | 14407/17688 [00:49<00:11, 293.02it/s]
 82%|████████████████████████████████████████▉         | 14442/17688 [00:49<00:10, 306.67it/s]
 82%|████████████████████████████████████████▉         | 14474/17688 [00:49<00:11, 279.15it/s]
 82%|█████████████████████████████████████████         | 14503/17688 [00:49<00:11, 271.46it/s]
 82%|█████████████████████████████████████████▏        | 14536/17688 [00:50<00:11, 286.18it/s]
 82%|█████████████████████████████████████████▏        | 14572/17688 [00:50<00:10, 304.37it/s]
 83%|█████████████████████████████████████████▎        | 14604/17688 [00:50<00:10, 306.48it/s]
 83%|█████████████████████████████████████████▍        | 14636/17688 [00:50<00:10, 288.06it/s]
 83%|█████████████████████████████████████████▍        | 14669/17688 [00:50<00:10, 298.08it/s]
 83%|█████████████████████████████████████████▌        | 14701/17688 [00:50<00:09, 302.84it/s]
 83%|█████████████████████████████████████████▋        | 14732/17688 [00:50<00:10, 282.71it/s]
 83%|█████████████████████████████████████████▋        | 14763/17688 [00:50<00:10, 288.17it/s]
 84%|█████████████████████████████████████████▊        | 14793/17688 [00:50<00:10, 279.64it/s]
 84%|█████████████████████████████████████████▉        | 14823/17688 [00:51<00:10, 284.86it/s]
 84%|██████████████████████████████████████████        | 14857/17688 [00:51<00:09, 297.27it/s]
 84%|██████████████████████████████████████████        | 14890/17688 [00:51<00:09, 303.26it/s]
 84%|██████████████████████████████████████████▏       | 14921/17688 [00:51<00:09, 295.89it/s]
 85%|██████████████████████████████████████████▎       | 14951/17688 [00:51<00:09, 293.01it/s]
 85%|██████████████████████████████████████████▎
```

```
| 14981/17688 [00:51<00:09, 290.17it/s]
85%|████████████████████████████████████████████▋
| 15011/17688 [00:51<00:09, 280.17it/s]
85%|████████████████████████████████████████████▊
| 15042/17688 [00:51<00:09, 287.89it/s]
85%|████████████████████████████████████████████▊
| 15071/17688 [00:51<00:09, 278.81it/s]
85%|█████████████████████████████████████████████
| 15100/17688 [00:51<00:09, 280.66it/s]
86%|█████████████████████████████████████████████▏
| 15133/17688 [00:52<00:08, 293.27it/s]
86%|█████████████████████████████████████████████▎
| 15163/17688 [00:52<00:08, 294.62it/s]
86%|█████████████████████████████████████████████▍
| 15194/17688 [00:52<00:08, 297.58it/s]
86%|█████████████████████████████████████████████
| 15224/17688 [00:52<00:08, 297.65it/s]
86%|█████████████████████████████████████████████▋
| 15255/17688 [00:52<00:08, 299.74it/s]
86%|█████████████████████████████████████████████▊
| 15286/17688 [00:52<00:08, 298.62it/s]
87%|█████████████████████████████████████████████▉
| 15317/17688 [00:52<00:07, 298.70it/s]
87%|██████████████████████████████████████████████
| 15350/17688 [00:52<00:07, 305.96it/s]
87%|██████████████████████████████████████████████▏
| 15381/17688 [00:52<00:07, 290.19it/s]
87%|██████████████████████████████████████████████▎
| 15411/17688 [00:53<00:07, 289.07it/s]
87%|██████████████████████████████████████████████▍
| 15442/17688 [00:53<00:07, 294.43it/s]
87%|██████████████████████████████████████████████▍
| 15472/17688 [00:53<00:08, 274.42it/s]
88%|██████████████████████████████████████████████▋
| 15501/17688 [00:53<00:07, 275.95it/s]
88%|██████████████████████████████████████████████▊
| 15532/17688 [00:53<00:07, 284.00it/s]
88%|██████████████████████████████████████████████▊
| 15561/17688 [00:53<00:07, 281.02it/s]
88%|███████████████████████████████████████████████
| 15597/17688 [00:53<00:06, 300.27it/s]
88%|███████████████████████████████████████████████▏
| 15628/17688 [00:53<00:06, 301.59it/s]
89%|███████████████████████████████████████████████▎
| 15663/17688 [00:53<00:06, 311.52it/s]
89%|███████████████████████████████████████████████▎
| 15695/17688 [00:53<00:06, 311.52it/s]
89%|███████████████████████████████████████████████▌
| 15727/17688 [00:54<00:06, 292.76it/s]
89%|███████████████████████████████████████████████▋
| 15757/17688 [00:54<00:06, 292.54it/s]
89%|███████████████████████████████████████████████▊
| 15787/17688 [00:54<00:06, 286.54it/s]
89%|███████████████████████████████████████████████▊
| 15816/17688 [00:54<00:06, 280.31it/s]
90%|████████████████████████████████████████████████
| 15851/17688 [00:54<00:06, 297.56it/s]
```

```
90%|█████████ | 15882/17688 [00:54<00:06, 296.25it/s]
90%|█████████ | 15912/17688 [00:54<00:06, 290.68it/s]
90%|█████████ | 15945/17688 [00:54<00:05, 300.87it/s]
90%|█████████ | 15976/17688 [00:54<00:05, 299.40it/s]
90%|█████████ | 16007/17688 [00:55<00:05, 299.24it/s]
91%|█████████ | 16038/17688 [00:55<00:05, 295.72it/s]
91%|█████████ | 16068/17688 [00:55<00:05, 285.38it/s]
91%|█████████ | 16099/17688 [00:55<00:05, 289.30it/s]
91%|█████████ | 16129/17688 [00:55<00:05, 281.96it/s]
91%|█████████▏| 16158/17688 [00:55<00:05, 274.09it/s]
92%|█████████▏| 16188/17688 [00:55<00:05, 277.69it/s]
92%|█████████▏| 16217/17688 [00:55<00:05, 280.68it/s]
92%|█████████▏| 16251/17688 [00:55<00:04, 291.82it/s]
92%|█████████▏| 16283/17688 [00:56<00:04, 297.46it/s]
92%|█████████▏| 16313/17688 [00:56<00:04, 294.08it/s]
92%|█████████▏| 16343/17688 [00:56<00:04, 278.78it/s]
93%|█████████▎| 16372/17688 [00:56<00:04, 267.47it/s]
93%|█████████▎| 16405/17688 [00:56<00:04, 282.33it/s]
93%|█████████▎| 16434/17688 [00:56<00:04, 279.88it/s]
93%|█████████▎| 16463/17688 [00:56<00:04, 281.42it/s]
93%|█████████▎| 16492/17688 [00:56<00:04, 280.87it/s]
93%|█████████▎| 16524/17688 [00:56<00:04, 290.98it/s]
94%|█████████▎| 16555/17688 [00:56<00:03, 295.82it/s]
94%|█████████▎| 16585/17688 [00:57<00:03, 286.26it/s]
94%|█████████▍| 16614/17688 [00:57<00:03, 271.47it/s]
94%|█████████▍| 16642/17688 [00:57<00:03, 270.25it/s]
94%|█████████▍| 16670/17688 [00:57<00:04, 247.97it/s]
94%|█████████▍| 16699/17688 [00:57<00:03, 257.36it/s]
95%|█████████▍
```

```
| 16727/17688 [00:57<00:03, 261.74it/s]
 95%|█████████████████████████████████████████████████
| 16757/17688 [00:57<00:03, 266.56it/s]
 95%|█████████████████████████████████████████████████▏
| 16790/17688 [00:57<00:03, 281.62it/s]
 95%|█████████████████████████████████████████████████▎
| 16824/17688 [00:57<00:02, 294.80it/s]
 95%|█████████████████████████████████████████████████▍
| 16859/17688 [00:58<00:02, 308.85it/s]
 96%|█████████████████████████████████████████████████▊
| 16893/17688 [00:58<00:02, 310.86it/s]
 96%|██████████████████████████████████████████████████
| 16927/17688 [00:58<00:02, 317.52it/s]
 96%|██████████████████████████████████████████████████
| 16960/17688 [00:58<00:02, 316.80it/s]
 96%|██████████████████████████████████████████████████
| 16993/17688 [00:58<00:02, 319.97it/s]
 96%|██████████████████████████████████████████████████
▏   | 17026/17688 [00:58<00:02, 317.58it/s]
 96%|██████████████████████████████████████████████████
▎   | 17062/17688 [00:58<00:01, 325.01it/s]
 97%|██████████████████████████████████████████████████
▍   | 17096/17688 [00:58<00:01, 326.78it/s]
 97%|██████████████████████████████████████████████████
▊   | 17130/17688 [00:58<00:01, 328.98it/s]
 97%|██████████████████████████████████████████████████
▉   | 17163/17688 [00:59<00:01, 316.31it/s]
 97%|██████████████████████████████████████████████████
█   | 17195/17688 [00:59<00:01, 310.29it/s]
 97%|██████████████████████████████████████████████████
▊  | 17227/17688 [00:59<00:01, 304.47it/s]
 98%|██████████████████████████████████████████████████
▊   | 17260/17688 [00:59<00:01, 310.18it/s]
 98%|██████████████████████████████████████████████████
█   | 17292/17688 [00:59<00:01, 299.28it/s]
 98%|██████████████████████████████████████████████████
█▏  | 17323/17688 [00:59<00:01, 277.52it/s]
 98%|██████████████████████████████████████████████████
█  | 17356/17688 [00:59<00:01, 289.34it/s]
 98%|██████████████████████████████████████████████████
█▍  | 17386/17688 [00:59<00:01, 279.62it/s]
 98%|██████████████████████████████████████████████████
█▋  | 17415/17688 [00:59<00:00, 278.00it/s]
 99%|██████████████████████████████████████████████████
█▊  | 17450/17688 [00:59<00:00, 295.75it/s]
 99%|██████████████████████████████████████████████████
█▍ | 17481/17688 [01:00<00:00, 295.83it/s]
 99%|██████████████████████████████████████████████████
█▌  | 17511/17688 [01:00<00:00, 294.68it/s]
 99%|██████████████████████████████████████████████████
█▉  | 17541/17688 [01:00<00:00, 281.51it/s]
 99%|██████████████████████████████████████████████████
█▉  | 17570/17688 [01:00<00:00, 278.51it/s]
 99%|██████████████████████████████████████████████████
█▊ | 17599/17688 [01:00<00:00, 276.45it/s]
100%|██████████████████████████████████████████████████
██▍ | 17630/17688 [01:00<00:00, 284.37it/s]
```

```
100%|
| 17659/17688 [01:00<00:00, 284.58it/s]
100%|
| 17688/17688 [01:00<00:00, 290.69it/s]

17688
300
```

In [118]:
```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_test = []; # the avg-w2v for each sentence/review is stored
 in this list
for sentence in tqdm(X_test['essay']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/revie
w
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf v
alue((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split
())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_test.append(vector)

print(len(tfidf_w2v_vectors_test))
print(len(tfidf_w2v_vectors_test[0]))
```

```
  0%|
| 0/26400 [00:00<?, ?it/s]
  0%|
| 27/26400 [00:00<01:39, 265.44it/s]
  0%|
| 55/26400 [00:00<01:38, 267.54it/s]
  0%|
| 89/26400 [00:00<01:32, 285.29it/s]
  0%|
| 118/26400 [00:00<01:32, 285.22it/s]
  1%|
| 145/26400 [00:00<01:34, 278.12it/s]
  1%|
| 172/26400 [00:00<01:35, 274.18it/s]
  1%|
| 207/26400 [00:00<01:30, 290.52it/s]
  1%|
| 235/26400 [00:00<01:31, 286.64it/s]
  1%|
| 265/26400 [00:00<01:31, 286.59it/s]
  1%|
| 300/26400 [00:01<01:26, 301.71it/s]
  1%|
| 330/26400 [00:01<01:30, 287.60it/s]
  1%|
| 362/26400 [00:01<01:28, 293.57it/s]
  1%|
| 395/26400 [00:01<01:26, 299.72it/s]
  2%|
| 425/26400 [00:01<01:28, 293.89it/s]
  2%|
| 455/26400 [00:01<01:30, 288.27it/s]
  2%|
| 484/26400 [00:01<01:31, 283.94it/s]
  2%|
| 515/26400 [00:01<01:29, 288.26it/s]
  2%|
| 546/26400 [00:01<01:29, 288.93it/s]
  2%|
| 575/26400 [00:01<01:31, 281.90it/s]
  2%|
| 609/26400 [00:02<01:27, 295.79it/s]
  2%|
| 639/26400 [00:02<01:27, 294.66it/s]
  3%|
| 669/26400 [00:02<01:28, 289.62it/s]
  3%|
| 699/26400 [00:02<01:33, 275.20it/s]
  3%|
| 733/26400 [00:02<01:28, 290.60it/s]
  3%|
| 763/26400 [00:02<01:27, 292.73it/s]
  3%|
| 793/26400 [00:02<01:29, 285.84it/s]
  3%|
| 822/26400 [00:02<01:36, 263.85it/s]
  3%|
```

```
| 852/26400 [00:02<01:33, 272.46it/s]
  3%|██▌
| 881/26400 [00:03<01:32, 276.12it/s]
  3%|███
| 909/26400 [00:03<01:32, 275.05it/s]
  4%|███
| 937/26400 [00:03<01:34, 269.56it/s]
  4%|███▍
| 967/26400 [00:03<01:32, 275.93it/s]
  4%|███▌
| 996/26400 [00:03<01:31, 277.81it/s]
  4%|███▌
| 1024/26400 [00:03<01:35, 266.77it/s]
  4%|███
| 1053/26400 [00:03<01:32, 272.78it/s]
  4%|███
| 1081/26400 [00:03<01:32, 272.72it/s]
  4%|███▎
| 1110/26400 [00:03<01:31, 277.08it/s]
  4%|███▍
| 1142/26400 [00:04<01:27, 287.37it/s]
  4%|███▍
| 1171/26400 [00:04<01:30, 277.64it/s]
  5%|███▊
| 1207/26400 [00:04<01:25, 293.91it/s]
  5%|███▍
| 1237/26400 [00:04<01:27, 286.66it/s]
  5%|████▏
| 1268/26400 [00:04<01:26, 290.19it/s]
  5%|████▏
| 1298/26400 [00:04<01:33, 269.66it/s]
  5%|████▍
| 1326/26400 [00:04<01:32, 271.31it/s]
  5%|████▋
| 1357/26400 [00:04<01:29, 279.78it/s]
  5%|████
| 1394/26400 [00:04<01:23, 299.14it/s]
  5%|████
| 1425/26400 [00:04<01:26, 289.05it/s]
  6%|████▎
| 1455/26400 [00:05<01:28, 281.79it/s]
  6%|████▌
| 1484/26400 [00:05<01:27, 283.57it/s]
  6%|████▌
| 1513/26400 [00:05<01:28, 282.39it/s]
  6%|████▊
| 1542/26400 [00:05<01:27, 284.02it/s]
  6%|████▌
| 1571/26400 [00:05<01:27, 282.68it/s]
  6%|████▌
| 1601/26400 [00:05<01:26, 286.24it/s]
  6%|█████▏
| 1630/26400 [00:05<01:26, 286.73it/s]
  6%|█████▍
| 1659/26400 [00:05<01:29, 277.23it/s]
  6%|█████▍
| 1687/26400 [00:05<01:30, 274.20it/s]
```

```
  6%|█████▌
| 1715/26400 [00:06<01:30, 273.71it/s]
  7%|█████
| 1746/26400 [00:06<01:27, 281.56it/s]
  7%|█████▎
| 1783/26400 [00:06<01:21, 302.03it/s]
  7%|█████▎
| 1814/26400 [00:06<01:22, 297.62it/s]
  7%|█████▌
| 1850/26400 [00:06<01:18, 312.52it/s]
  7%|█████▋
| 1882/26400 [00:06<01:18, 311.31it/s]
  7%|█████▌
| 1914/26400 [00:06<01:19, 308.67it/s]
  7%|█████▌
| 1946/26400 [00:06<01:21, 299.11it/s]
  7%|██████
| 1977/26400 [00:06<01:23, 293.14it/s]
  8%|██████▎
| 2007/26400 [00:06<01:24, 289.39it/s]
  8%|██████▎
| 2037/26400 [00:07<01:23, 290.22it/s]
  8%|██████▌
| 2067/26400 [00:07<01:31, 266.78it/s]
  8%|██████
| 2095/26400 [00:07<01:33, 259.56it/s]
  8%|██████
| 2125/26400 [00:07<01:30, 268.52it/s]
  8%|██████▏
| 2154/26400 [00:07<01:30, 267.25it/s]
  8%|██████▍
| 2181/26400 [00:07<01:30, 266.70it/s]
  8%|██████▍
| 2212/26400 [00:07<01:27, 277.07it/s]
  8%|██████▋
| 2242/26400 [00:07<01:25, 282.99it/s]
  9%|██████▌
| 2275/26400 [00:07<01:21, 295.04it/s]
  9%|███████
| 2305/26400 [00:08<01:22, 291.57it/s]
  9%|███████
| 2335/26400 [00:08<01:22, 292.56it/s]
  9%|███████▎
| 2365/26400 [00:08<01:24, 283.32it/s]
  9%|███████▌
| 2394/26400 [00:08<01:25, 279.75it/s]
  9%|███████▌
| 2424/26400 [00:08<01:24, 284.13it/s]
  9%|███████
| 2454/26400 [00:08<01:23, 288.10it/s]
  9%|███████▏
| 2488/26400 [00:08<01:19, 300.54it/s]
 10%|███████▍
| 2519/26400 [00:08<01:20, 294.92it/s]
 10%|███████▍
| 2549/26400 [00:08<01:27, 272.47it/s]
 10%|███████▋
```

```
| 2579/26400 [00:09<01:25, 279.59it/s]
 10%|████████
| 2609/26400 [00:09<01:23, 283.24it/s]
 10%|████████
| 2638/26400 [00:09<01:23, 283.79it/s]
 10%|████████▋
| 2667/26400 [00:09<01:24, 281.69it/s]
 10%|████████▉
| 2697/26400 [00:09<01:23, 285.53it/s]
 10%|████████▉
| 2726/26400 [00:09<01:23, 283.71it/s]
 10%|█████████
| 2755/26400 [00:09<01:23, 282.47it/s]
 11%|████████▌
| 2787/26400 [00:09<01:22, 285.18it/s]
 11%|████████▌
| 2817/26400 [00:09<01:21, 288.01it/s]
 11%|████████▊
| 2851/26400 [00:09<01:18, 299.70it/s]
 11%|█████████
| 2882/26400 [00:10<01:27, 269.12it/s]
 11%|█████████
| 2914/26400 [00:10<01:23, 281.31it/s]
 11%|█████████
| 2943/26400 [00:10<01:24, 276.00it/s]
 11%|█████████
| 2977/26400 [00:10<01:20, 291.22it/s]
 11%|█████████▊
| 3009/26400 [00:10<01:19, 294.57it/s]
 12%|██████████
| 3041/26400 [00:10<01:17, 300.29it/s]
 12%|██████████
| 3072/26400 [00:10<01:18, 299.00it/s]
 12%|██████████▏
| 3103/26400 [00:10<01:19, 291.40it/s]
 12%|█████████▋
| 3133/26400 [00:10<01:20, 287.41it/s]
 12%|█████████▋
| 3162/26400 [00:11<01:25, 272.21it/s]
 12%|█████████▊
| 3194/26400 [00:11<01:21, 284.42it/s]
 12%|██████████
| 3223/26400 [00:11<01:21, 283.79it/s]
 12%|██████████
| 3252/26400 [00:11<01:21, 285.01it/s]
 12%|██████████▎
| 3281/26400 [00:11<01:23, 278.47it/s]
 13%|██████████
| 3309/26400 [00:11<01:24, 273.42it/s]
 13%|██████████
| 3340/26400 [00:11<01:22, 280.61it/s]
 13%|██████████▊
| 3370/26400 [00:11<01:20, 285.56it/s]
 13%|███████████
| 3399/26400 [00:11<01:21, 283.74it/s]
 13%|███████████
| 3428/26400 [00:12<01:23, 275.27it/s]
```

```
 13%|████
| 3458/26400 [00:12<01:21, 281.67it/s]
 13%|████
| 3488/26400 [00:12<01:20, 284.70it/s]
 13%|████
| 3517/26400 [00:12<01:21, 280.69it/s]
 13%|████
| 3548/26400 [00:12<01:19, 287.49it/s]
 14%|████
| 3577/26400 [00:12<01:19, 286.76it/s]
 14%|████
| 3608/26400 [00:12<01:18, 291.11it/s]
 14%|████
| 3639/26400 [00:12<01:18, 291.74it/s]
 14%|████
| 3672/26400 [00:12<01:17, 293.62it/s]
 14%|████
| 3702/26400 [00:12<01:20, 280.83it/s]
 14%|████
| 3731/26400 [00:13<01:22, 274.11it/s]
 14%|████
| 3762/26400 [00:13<01:20, 282.63it/s]
 14%|████
| 3795/26400 [00:13<01:17, 292.43it/s]
 14%|████
| 3825/26400 [00:13<01:18, 286.47it/s]
 15%|████
| 3854/26400 [00:13<01:21, 277.85it/s]
 15%|████
| 3891/26400 [00:13<01:15, 299.03it/s]
 15%|████
| 3922/26400 [00:13<01:21, 277.37it/s]
 15%|████
| 3951/26400 [00:13<01:20, 280.45it/s]
 15%|████
| 3980/26400 [00:13<01:19, 282.64it/s]
 15%|████
| 4009/26400 [00:14<01:21, 273.76it/s]
 15%|████
| 4037/26400 [00:14<01:22, 270.25it/s]
 15%|████
| 4065/26400 [00:14<01:24, 265.56it/s]
 16%|████
| 4094/26400 [00:14<01:22, 271.89it/s]
 16%|████
| 4127/26400 [00:14<01:18, 282.11it/s]
 16%|████
| 4160/26400 [00:14<01:15, 294.37it/s]
 16%|████
| 4192/26400 [00:14<01:15, 295.18it/s]
 16%|████
| 4222/26400 [00:14<01:16, 289.98it/s]
 16%|████
| 4252/26400 [00:14<01:17, 286.44it/s]
 16%|████
| 4281/26400 [00:15<01:20, 275.46it/s]
 16%|████
```

```
| 4311/26400 [00:15<01:18, 281.02it/s]
16%|████████████▌
| 4346/26400 [00:15<01:14, 296.61it/s]
17%|████████████▌
| 4376/26400 [00:15<01:15, 291.79it/s]
17%|████████████▉
| 4406/26400 [00:15<01:14, 293.57it/s]
17%|█████████████
| 4436/26400 [00:15<01:15, 290.56it/s]
17%|█████████████
| 4471/26400 [00:15<01:11, 304.78it/s]
17%|█████████████▌
| 4502/26400 [00:15<01:14, 294.38it/s]
17%|████████████▉
| 4532/26400 [00:15<01:16, 286.95it/s]
17%|█████████████
| 4561/26400 [00:15<01:16, 285.54it/s]
17%|█████████████
| 4592/26400 [00:16<01:14, 291.04it/s]
18%|█████████████▍
| 4623/26400 [00:16<01:14, 294.18it/s]
18%|█████████████▋
| 4653/26400 [00:16<01:17, 281.98it/s]
18%|█████████████▋
| 4684/26400 [00:16<01:15, 287.64it/s]
18%|█████████████▍
| 4714/26400 [00:16<01:15, 288.11it/s]
18%|██████████████▏
| 4744/26400 [00:16<01:14, 290.12it/s]
18%|██████████████▎
| 4779/26400 [00:16<01:11, 303.65it/s]
18%|██████████████▎
| 4811/26400 [00:16<01:10, 307.72it/s]
18%|██████████████▌
| 4842/26400 [00:16<01:10, 305.92it/s]
18%|█████████████▉
| 4873/26400 [00:16<01:11, 302.88it/s]
19%|█████████████▉
| 4904/26400 [00:17<01:11, 299.92it/s]
19%|██████████████▏
| 4936/26400 [00:17<01:11, 299.07it/s]
19%|██████████████▍
| 4966/26400 [00:17<01:13, 290.05it/s]
19%|██████████████▋
| 4996/26400 [00:17<01:16, 278.53it/s]
19%|██████████████▋
| 5027/26400 [00:17<01:14, 285.12it/s]
19%|██████████████▍
| 5061/26400 [00:17<01:11, 299.05it/s]
19%|███████████████
| 5092/26400 [00:17<01:14, 285.02it/s]
19%|███████████████
| 5121/26400 [00:17<01:17, 275.32it/s]
20%|███████████████▎
| 5156/26400 [00:17<01:12, 293.61it/s]
20%|███████████████▌
| 5190/26400 [00:18<01:09, 303.09it/s]
```

```
20%|
| 5222/26400 [00:18<01:09, 306.44it/s]
20%|
| 5253/26400 [00:18<01:11, 293.78it/s]
20%|
| 5283/26400 [00:18<01:12, 290.71it/s]
20%|
| 5313/26400 [00:18<01:13, 286.94it/s]
20%|
| 5342/26400 [00:18<01:13, 286.38it/s]
20%|
| 5376/26400 [00:18<01:10, 299.23it/s]
20%|
| 5407/26400 [00:18<01:10, 295.71it/s]
21%|
| 5441/26400 [00:18<01:08, 306.30it/s]
21%|
| 5474/26400 [00:19<01:06, 312.39it/s]
21%|
| 5513/26400 [00:19<01:03, 330.77it/s]
21%|
| 5547/26400 [00:19<01:03, 329.87it/s]
21%|
| 5582/26400 [00:19<01:02, 334.00it/s]
21%|
| 5616/26400 [00:19<01:08, 304.53it/s]
21%|
| 5648/26400 [00:19<01:09, 300.56it/s]
22%|
| 5687/26400 [00:19<01:04, 320.59it/s]
22%|
| 5720/26400 [00:19<01:05, 314.39it/s]
22%|
| 5752/26400 [00:19<01:07, 304.59it/s]
22%|
| 5783/26400 [00:20<01:08, 301.09it/s]
22%|
| 5814/26400 [00:20<01:08, 298.68it/s]
22%|
| 5847/26400 [00:20<01:07, 305.97it/s]
22%|
| 5878/26400 [00:20<01:07, 304.70it/s]
22%|
| 5909/26400 [00:20<01:07, 302.03it/s]
22%|
| 5940/26400 [00:20<01:07, 301.09it/s]
23%|
| 5971/26400 [00:20<01:08, 297.83it/s]
23%|
| 6003/26400 [00:20<01:07, 300.96it/s]
23%|
| 6035/26400 [00:20<01:06, 304.91it/s]
23%|
| 6067/26400 [00:20<01:06, 306.86it/s]
23%|
| 6098/26400 [00:21<01:06, 304.42it/s]
23%|
```

```
| 6129/26400 [00:21<01:07, 298.37it/s]
 23%|
| 6160/26400 [00:21<01:07, 300.26it/s]
 23%|
| 6191/26400 [00:21<01:06, 302.47it/s]
 24%|
| 6222/26400 [00:21<01:09, 291.22it/s]
 24%|
| 6252/26400 [00:21<01:09, 289.78it/s]
 24%|
| 6285/26400 [00:21<01:07, 299.36it/s]
 24%|
| 6316/26400 [00:21<01:08, 294.96it/s]
 24%|
| 6346/26400 [00:21<01:07, 295.81it/s]
 24%|
| 6376/26400 [00:21<01:08, 292.95it/s]
 24%|
| 6406/26400 [00:22<01:13, 272.77it/s]
 24%|
| 6434/26400 [00:22<01:14, 267.26it/s]
 24%|
| 6465/26400 [00:22<01:12, 276.76it/s]
 25%|
| 6493/26400 [00:22<01:13, 269.92it/s]
 25%|
| 6521/26400 [00:22<01:14, 268.38it/s]
 25%|
| 6548/26400 [00:22<01:16, 258.30it/s]
 25%|
| 6583/26400 [00:22<01:10, 279.84it/s]
 25%|
| 6612/26400 [00:22<01:11, 277.35it/s]
 25%|
| 6643/26400 [00:22<01:09, 285.84it/s]
 25%|
| 6676/26400 [00:23<01:06, 297.20it/s]
 25%|
| 6707/26400 [00:23<01:08, 288.58it/s]
 26%|
| 6739/26400 [00:23<01:06, 295.10it/s]
 26%|
| 6769/26400 [00:23<01:07, 290.75it/s]
 26%|
| 6799/26400 [00:23<01:07, 288.63it/s]
 26%|
| 6829/26400 [00:23<01:07, 291.32it/s]
 26%|
| 6859/26400 [00:23<01:08, 283.31it/s]
 26%|
| 6888/26400 [00:23<01:10, 274.98it/s]
 26%|
| 6922/26400 [00:23<01:08, 285.32it/s]
 26%|
| 6953/26400 [00:24<01:06, 291.70it/s]
 26%|
| 6983/26400 [00:24<01:06, 293.51it/s]
```

```
 27%|████████████████████▏
| 7013/26400 [00:24<01:07, 288.85it/s]
 27%|████████████████████▎
| 7043/26400 [00:24<01:06, 289.79it/s]
 27%|████████████████████▎
| 7073/26400 [00:24<01:07, 287.14it/s]
 27%|████████████████████▍
| 7104/26400 [00:24<01:06, 292.20it/s]
 27%|████████████████████▌
| 7134/26400 [00:24<01:06, 291.31it/s]
 27%|████████████████████▋
| 7168/26400 [00:24<01:03, 300.57it/s]
 27%|████████████████████▋
| 7199/26400 [00:24<01:06, 290.80it/s]
 27%|████████████████████▊
| 7230/26400 [00:24<01:04, 295.68it/s]
 28%|████████████████████▉
| 7260/26400 [00:25<01:05, 290.32it/s]
 28%|████████████████████▉
| 7290/26400 [00:25<01:05, 291.68it/s]
 28%|█████████████████████
| 7320/26400 [00:25<01:07, 281.93it/s]
 28%|█████████████████████▏
| 7349/26400 [00:25<01:08, 276.42it/s]
 28%|█████████████████████▏
| 7381/26400 [00:25<01:06, 287.63it/s]
 28%|█████████████████████▎
| 7414/26400 [00:25<01:04, 295.36it/s]
 28%|█████████████████████▍
| 7447/26400 [00:25<01:02, 304.35it/s]
 28%|█████████████████████▌
| 7478/26400 [00:25<01:05, 290.79it/s]
 28%|█████████████████████▌
| 7508/26400 [00:25<01:06, 285.36it/s]
 29%|█████████████████████▋
| 7539/26400 [00:26<01:04, 290.91it/s]
 29%|█████████████████████▊
| 7569/26400 [00:26<01:05, 289.55it/s]
 29%|█████████████████████▉
| 7601/26400 [00:26<01:03, 297.47it/s]
 29%|█████████████████████▉
| 7631/26400 [00:26<01:05, 284.87it/s]
 29%|██████████████████████
| 7660/26400 [00:26<01:06, 280.02it/s]
 29%|██████████████████████▏
| 7692/26400 [00:26<01:05, 286.45it/s]
 29%|██████████████████████▏
| 7721/26400 [00:26<01:06, 279.44it/s]
 29%|██████████████████████▎
| 7750/26400 [00:26<01:06, 281.93it/s]
 29%|██████████████████████▍
| 7779/26400 [00:26<01:06, 278.00it/s]
 30%|██████████████████████▍
| 7807/26400 [00:27<01:07, 274.72it/s]
 30%|██████████████████████▌
| 7840/26400 [00:27<01:04, 287.95it/s]
 30%|██████████████████████▋
```

```
| 7870/26400 [00:27<01:06, 279.49it/s]
30%|███████████████████████▏
| 7899/26400 [00:27<01:08, 268.68it/s]
30%|███████████████████████▍
| 7934/26400 [00:27<01:04, 287.58it/s]
30%|███████████████████████▌
| 7964/26400 [00:27<01:03, 289.73it/s]
30%|████████████████████████
| 7994/26400 [00:27<01:03, 287.92it/s]
30%|████████████████████████
| 8024/26400 [00:27<01:04, 283.43it/s]
31%|████████████████████████▏
| 8054/26400 [00:27<01:03, 286.77it/s]
31%|████████████████████████▎
| 8083/26400 [00:27<01:03, 287.11it/s]
31%|████████████████████████▎
| 8117/26400 [00:28<01:00, 300.58it/s]
31%|████████████████████████▍
| 8148/26400 [00:28<01:02, 292.44it/s]
31%|████████████████████████▌
| 8179/26400 [00:28<01:02, 292.68it/s]
31%|████████████████████████▋
| 8209/26400 [00:28<01:03, 285.82it/s]
31%|████████████████████████▋
| 8241/26400 [00:28<01:01, 293.88it/s]
31%|████████████████████████▊
| 8271/26400 [00:28<01:02, 289.93it/s]
31%|████████████████████████▉
| 8302/26400 [00:28<01:01, 293.38it/s]
32%|████████████████████████▉
| 8333/26400 [00:28<01:00, 296.69it/s]
32%|█████████████████████████
| 8363/26400 [00:28<01:01, 295.29it/s]
32%|█████████████████████████▏
| 8393/26400 [00:29<01:06, 271.96it/s]
32%|█████████████████████████▏
| 8421/26400 [00:29<01:07, 265.20it/s]
32%|█████████████████████████▎
| 8452/26400 [00:29<01:04, 276.68it/s]
32%|█████████████████████████▍
| 8484/26400 [00:29<01:02, 286.28it/s]
32%|█████████████████████████▌
| 8514/26400 [00:29<01:02, 287.14it/s]
32%|█████████████████████████▌
| 8549/26400 [00:29<00:59, 299.83it/s]
32%|█████████████████████████▋
| 8580/26400 [00:29<01:01, 287.88it/s]
33%|█████████████████████████▊
| 8611/26400 [00:29<01:01, 291.10it/s]
33%|█████████████████████████▉
| 8641/26400 [00:29<01:00, 293.09it/s]
33%|█████████████████████████▉
| 8671/26400 [00:29<01:01, 289.39it/s]
33%|██████████████████████████
| 8702/26400 [00:30<01:00, 293.81it/s]
33%|██████████████████████████▏
| 8735/26400 [00:30<00:58, 303.21it/s]
```

```
 33%|█████████████████████████▏| 8766/26400 [00:30<00:58, 301.02it/s]
 33%|█████████████████████████▍| 8797/26400 [00:30<00:58, 302.13it/s]
 33%|█████████████████████████▌| 8828/26400 [00:30<01:02, 279.23it/s]
 34%|█████████████████████████▌| 8857/26400 [00:30<01:03, 276.16it/s]
 34%|█████████████████████████▊| 8885/26400 [00:30<01:04, 272.67it/s]
 34%|█████████████████████████▉| 8913/26400 [00:30<01:04, 272.64it/s]
 34%|█████████████████████████▉| 8943/26400 [00:30<01:02, 278.96it/s]
 34%|█████████████████████████▉| 8976/26400 [00:31<00:59, 290.43it/s]
 34%|██████████████████████████| 9006/26400 [00:31<01:03, 272.71it/s]
 34%|██████████████████████████| 9038/26400 [00:31<01:00, 284.81it/s]
 34%|██████████████████████████| 9067/26400 [00:31<01:01, 281.56it/s]
 34%|██████████████████████████▏| 9102/26400 [00:31<00:58, 296.29it/s]
 35%|██████████████████████████▍| 9133/26400 [00:31<00:59, 287.98it/s]
 35%|██████████████████████████▋| 9163/26400 [00:31<01:00, 284.27it/s]
 35%|██████████████████████████▋| 9196/26400 [00:31<00:58, 295.22it/s]
 35%|███████████████████████████| 9226/26400 [00:31<00:59, 287.51it/s]
 35%|██████████████████████████▉| 9255/26400 [00:32<00:59, 285.92it/s]
 35%|██████████████████████████▉| 9284/26400 [00:32<01:02, 275.92it/s]
 35%|███████████████████████████| 9315/26400 [00:32<01:00, 283.96it/s]
 35%|███████████████████████████▏| 9344/26400 [00:32<01:04, 263.44it/s]
 35%|███████████████████████████▏| 9371/26400 [00:32<01:04, 263.26it/s]
 36%|███████████████████████████| 9401/26400 [00:32<01:02, 271.26it/s]
 36%|███████████████████████████▏| 9429/26400 [00:32<01:02, 272.46it/s]
 36%|███████████████████████████▏| 9457/26400 [00:32<01:04, 262.56it/s]
 36%|███████████████████████████▍| 9489/26400 [00:32<01:01, 276.26it/s]
 36%|███████████████████████████▋| 9517/26400 [00:33<01:05, 257.01it/s]
 36%|███████████████████████████▋| 9548/26400 [00:33<01:02, 269.68it/s]
 36%|████████████████████████████| 9576/26400 [00:33<01:05, 256.45it/s]
 36%|████████████████████████████
```

```
| 9608/26400 [00:33<01:01, 272.19it/s]
 36%|
| 9636/26400 [00:33<01:02, 269.95it/s]
 37%|
| 9669/26400 [00:33<00:58, 284.26it/s]
 37%|
| 9698/26400 [00:33<00:58, 284.51it/s]
 37%|
| 9731/26400 [00:33<00:56, 295.40it/s]
 37%|
| 9761/26400 [00:33<00:56, 294.39it/s]
 37%|
| 9791/26400 [00:33<00:56, 294.54it/s]
 37%|
| 9821/26400 [00:34<00:56, 294.65it/s]
 37%|
| 9853/26400 [00:34<00:55, 297.84it/s]
 37%|
| 9886/26400 [00:34<00:53, 306.19it/s]
 38%|
| 9917/26400 [00:34<00:54, 304.85it/s]
 38%|
| 9948/26400 [00:34<00:55, 297.81it/s]
 38%|
| 9978/26400 [00:34<00:59, 277.23it/s]
 38%|
| 10009/26400 [00:34<00:57, 285.73it/s]
 38%|
| 10040/26400 [00:34<00:57, 284.77it/s]
 38%|
| 10069/26400 [00:34<00:57, 284.86it/s]
 38%|
| 10100/26400 [00:35<00:56, 290.55it/s]
 38%|
| 10132/26400 [00:35<00:54, 297.35it/s]
 38%|
| 10162/26400 [00:35<00:54, 297.50it/s]
 39%|
| 10192/26400 [00:35<00:55, 293.25it/s]
 39%|
| 10222/26400 [00:35<00:57, 281.38it/s]
 39%|
| 10255/26400 [00:35<00:55, 291.49it/s]
 39%|
| 10285/26400 [00:35<00:56, 287.47it/s]
 39%|
| 10320/26400 [00:35<00:53, 301.61it/s]
 39%|
| 10351/26400 [00:35<00:54, 293.13it/s]
 39%|
| 10383/26400 [00:35<00:53, 296.75it/s]
 39%|
| 10414/26400 [00:36<00:53, 299.97it/s]
 40%|
| 10445/26400 [00:36<00:53, 298.78it/s]
 40%|
| 10475/26400 [00:36<00:54, 292.40it/s]
```

```
 40%|
| 10505/26400 [00:36<00:55, 285.63it/s]
 40%|
| 10534/26400 [00:36<00:56, 281.32it/s]
 40%|
| 10563/26400 [00:36<00:55, 283.26it/s]
 40%|
| 10592/26400 [00:36<00:57, 274.17it/s]
 40%|
| 10621/26400 [00:36<00:56, 278.15it/s]
 40%|
| 10652/26400 [00:36<00:55, 285.62it/s]
 40%|
| 10681/26400 [00:37<00:54, 286.30it/s]
 41%|
| 10710/26400 [00:37<00:54, 285.92it/s]
 41%|
| 10743/26400 [00:37<00:52, 296.48it/s]
 41%|
| 10773/26400 [00:37<00:56, 274.91it/s]
 41%|
| 10801/26400 [00:37<00:56, 274.20it/s]
 41%|
| 10829/26400 [00:37<00:58, 268.22it/s]
 41%|
| 10860/26400 [00:37<00:56, 277.48it/s]
 41%|
| 10888/26400 [00:37<00:57, 271.19it/s]
 41%|
| 10916/26400 [00:37<00:56, 272.39it/s]
 41%|
| 10949/26400 [00:37<00:53, 286.15it/s]
 42%|
| 10979/26400 [00:38<00:54, 283.82it/s]
 42%|
| 11010/26400 [00:38<00:52, 290.60it/s]
 42%|
| 11041/26400 [00:38<00:52, 294.70it/s]
 42%|
| 11073/26400 [00:38<00:51, 297.88it/s]
 42%|
| 11105/26400 [00:38<00:50, 301.84it/s]
 42%|
| 11138/26400 [00:38<00:49, 306.56it/s]
 42%|
| 11169/26400 [00:38<00:50, 300.68it/s]
 42%|
| 11200/26400 [00:38<00:52, 290.87it/s]
 43%|
| 11233/26400 [00:38<00:50, 300.99it/s]
 43%|
| 11264/26400 [00:39<00:53, 283.90it/s]
 43%|
| 11298/26400 [00:39<00:50, 298.11it/s]
 43%|
| 11329/26400 [00:39<00:50, 300.06it/s]
 43%|
```

```
| 11362/26400 [00:39<00:48, 306.97it/s]
 43%|
| 11393/26400 [00:39<00:49, 303.61it/s]
 43%|
| 11424/26400 [00:39<00:51, 290.32it/s]
 43%|
| 11454/26400 [00:39<00:51, 291.68it/s]
 44%|
| 11484/26400 [00:39<00:52, 284.34it/s]
 44%|
| 11516/26400 [00:39<00:50, 291.97it/s]
 44%|
| 11546/26400 [00:40<00:52, 283.74it/s]
 44%|
| 11575/26400 [00:40<00:53, 279.23it/s]
 44%|
| 11604/26400 [00:40<00:53, 278.53it/s]
 44%|
| 11635/26400 [00:40<00:51, 285.12it/s]
 44%|
| 11664/26400 [00:40<00:52, 281.79it/s]
 44%|
| 11693/26400 [00:40<00:53, 275.53it/s]
 44%|
| 11721/26400 [00:40<00:53, 273.83it/s]
 45%|
| 11754/26400 [00:40<00:50, 287.27it/s]
 45%|
| 11783/26400 [00:40<00:51, 284.92it/s]
 45%|
| 11812/26400 [00:40<00:52, 276.83it/s]
 45%|
| 11846/26400 [00:41<00:50, 289.63it/s]
 45%|
| 11879/26400 [00:41<00:48, 297.64it/s]
 45%|
| 11910/26400 [00:41<00:48, 299.73it/s]
 45%|
| 11941/26400 [00:41<00:49, 289.44it/s]
 45%|
| 11974/26400 [00:41<00:48, 299.92it/s]
 45%|
| 12005/26400 [00:41<00:49, 291.18it/s]
 46%|
| 12035/26400 [00:41<00:50, 286.45it/s]
 46%|
| 12064/26400 [00:41<00:52, 270.85it/s]
 46%|
| 12092/26400 [00:41<00:53, 266.72it/s]
 46%|
| 12128/26400 [00:42<00:49, 288.00it/s]
 46%|
| 12158/26400 [00:42<00:52, 271.21it/s]
 46%|
| 12190/26400 [00:42<00:50, 283.66it/s]
 46%|
| 12219/26400 [00:42<00:51, 275.21it/s]
```

```
46%|
| 12247/26400 [00:42<00:52, 269.67it/s]
47%|
| 12280/26400 [00:42<00:49, 284.78it/s]
47%|
| 12309/26400 [00:42<00:51, 275.95it/s]
47%|
| 12341/26400 [00:42<00:49, 286.48it/s]
47%|
| 12371/26400 [00:42<00:49, 281.67it/s]
47%|
| 12400/26400 [00:43<00:53, 262.05it/s]
47%|
| 12428/26400 [00:43<00:52, 265.87it/s]
47%|
| 12460/26400 [00:43<00:50, 278.09it/s]
47%|
| 12489/26400 [00:43<00:49, 280.15it/s]
47%|
| 12519/26400 [00:43<00:48, 285.23it/s]
48%|
| 12551/26400 [00:43<00:47, 293.44it/s]
48%|
| 12581/26400 [00:43<00:47, 290.47it/s]
48%|
| 12612/26400 [00:43<00:47, 292.94it/s]
48%|
| 12648/26400 [00:43<00:44, 309.70it/s]
48%|
| 12680/26400 [00:43<00:48, 285.40it/s]
48%|
| 12710/26400 [00:44<00:49, 279.36it/s]
48%|
| 12739/26400 [00:44<00:50, 268.59it/s]
48%|
| 12768/26400 [00:44<00:49, 273.33it/s]
48%|
| 12798/26400 [00:44<00:48, 280.24it/s]
49%|
| 12831/26400 [00:44<00:46, 289.87it/s]
49%|
| 12862/26400 [00:44<00:46, 290.87it/s]
49%|
| 12892/26400 [00:44<00:46, 292.07it/s]
49%|
| 12922/26400 [00:44<00:47, 286.22it/s]
49%|
| 12951/26400 [00:44<00:48, 276.89it/s]
49%|
| 12983/26400 [00:45<00:46, 287.99it/s]
49%|
| 13013/26400 [00:45<00:46, 289.19it/s]
49%|
| 13043/26400 [00:45<00:46, 287.54it/s]
50%|
| 13072/26400 [00:45<00:46, 286.80it/s]
50%|
```

```
| 13101/26400 [00:45<00:48, 274.11it/s]
 50%|
| 13131/26400 [00:45<00:47, 280.84it/s]
 50%|
| 13160/26400 [00:45<00:48, 273.35it/s]
 50%|
| 13190/26400 [00:45<00:47, 275.63it/s]
 50%|
| 13218/26400 [00:45<00:47, 275.52it/s]
 50%|
| 13250/26400 [00:46<00:45, 286.17it/s]
 50%|
| 13282/26400 [00:46<00:44, 293.33it/s]
 50%|
| 13317/26400 [00:46<00:42, 307.71it/s]
 51%|
| 13351/26400 [00:46<00:41, 314.34it/s]
 51%|
| 13383/26400 [00:46<00:42, 309.85it/s]
 51%|
| 13415/26400 [00:46<00:43, 297.40it/s]
 51%|
| 13445/26400 [00:46<00:44, 293.18it/s]
 51%|
| 13475/26400 [00:46<00:45, 283.72it/s]
 51%|
| 13504/26400 [00:46<00:48, 268.39it/s]
 51%|
| 13534/26400 [00:46<00:46, 275.83it/s]
 51%|
| 13563/26400 [00:47<00:46, 275.37it/s]
 51%|
| 13595/26400 [00:47<00:45, 284.54it/s]
 52%|
| 13624/26400 [00:47<00:45, 278.17it/s]
 52%|
| 13654/26400 [00:47<00:44, 283.79it/s]
 52%|
| 13685/26400 [00:47<00:43, 290.57it/s]
 52%|
| 13715/26400 [00:47<00:43, 291.86it/s]
 52%|
| 13749/26400 [00:47<00:41, 304.21it/s]
 52%|
| 13781/26400 [00:47<00:41, 307.24it/s]
 52%|
| 13812/26400 [00:47<00:42, 293.47it/s]
 52%|
| 13842/26400 [00:48<00:43, 287.98it/s]
 53%|
| 13871/26400 [00:48<00:43, 287.96it/s]
 53%|
| 13903/26400 [00:48<00:42, 295.46it/s]
 53%|
| 13935/26400 [00:48<00:41, 299.26it/s]
 53%|
| 13966/26400 [00:48<00:42, 294.05it/s]
```

```
 53%|
| 13996/26400 [00:48<00:43, 288.38it/s]
 53%|
| 14026/26400 [00:48<00:43, 282.93it/s]
 53%|
| 14055/26400 [00:48<00:43, 284.40it/s]
 53%|
| 14084/26400 [00:48<00:43, 280.49it/s]
 53%|
| 14115/26400 [00:48<00:43, 284.98it/s]
 54%|
| 14146/26400 [00:49<00:42, 291.45it/s]
 54%|
| 14176/26400 [00:49<00:42, 289.94it/s]
 54%|
| 14206/26400 [00:49<00:42, 287.24it/s]
 54%|
| 14235/26400 [00:49<00:42, 284.90it/s]
 54%|
| 14266/26400 [00:49<00:41, 290.57it/s]
 54%|
| 14296/26400 [00:49<00:42, 286.03it/s]
 54%|
| 14325/26400 [00:49<00:42, 285.74it/s]
 54%|
| 14361/26400 [00:49<00:40, 297.27it/s]
 55%|
| 14391/26400 [00:49<00:42, 285.58it/s]
 55%|
| 14422/26400 [00:50<00:41, 290.25it/s]
 55%|
| 14452/26400 [00:50<00:41, 286.63it/s]
 55%|
| 14482/26400 [00:50<00:41, 285.77it/s]
 55%|
| 14511/26400 [00:50<00:42, 278.98it/s]
 55%|
| 14539/26400 [00:50<00:42, 276.21it/s]
 55%|
| 14575/26400 [00:50<00:39, 296.41it/s]
 55%|
| 14606/26400 [00:50<00:39, 298.00it/s]
 55%|
| 14637/26400 [00:50<00:39, 298.26it/s]
 56%|
| 14668/26400 [00:50<00:40, 289.28it/s]
 56%|
| 14698/26400 [00:50<00:40, 287.61it/s]
 56%|
| 14727/26400 [00:51<00:40, 285.15it/s]
 56%|
| 14756/26400 [00:51<00:40, 285.97it/s]
 56%|
| 14785/26400 [00:51<00:41, 279.12it/s]
 56%|
| 14818/26400 [00:51<00:39, 292.09it/s]
 56%|
```

```
| 14848/26400 [00:51<00:40, 283.81it/s]
 56%|█████████████████████████████████████████▋
| 14879/26400 [00:51<00:40, 284.97it/s]
 56%|█████████████████████████████████████████▋
| 14912/26400 [00:51<00:39, 293.39it/s]
 57%|█████████████████████████████████████████▋
| 14944/26400 [00:51<00:38, 296.97it/s]
 57%|██████████████████████████████████████████
| 14974/26400 [00:51<00:40, 285.38it/s]
 57%|█████████████████████████████████████████▉
| 15006/26400 [00:52<00:38, 292.75it/s]
 57%|█████████████████████████████████████████▉
| 15037/26400 [00:52<00:38, 297.09it/s]
 57%|█████████████████████████████████████████▉
| 15067/26400 [00:52<00:39, 288.75it/s]
 57%|█████████████████████████████████████████▉
| 15097/26400 [00:52<00:39, 289.73it/s]
 57%|█████████████████████████████████████████▉
| 15127/26400 [00:52<00:39, 284.65it/s]
 57%|██████████████████████████████████████████
| 15158/26400 [00:52<00:38, 290.39it/s]
 58%|██████████████████████████████████████████▏
| 15188/26400 [00:52<00:39, 280.31it/s]
 58%|██████████████████████████████████████████▏
| 15219/26400 [00:52<00:39, 286.42it/s]
 58%|██████████████████████████████████████████▏
| 15248/26400 [00:52<00:39, 284.34it/s]
 58%|██████████████████████████████████████████▎
| 15277/26400 [00:53<00:39, 280.44it/s]
 58%|██████████████████████████████████████████▎
| 15306/26400 [00:53<00:39, 280.19it/s]
 58%|██████████████████████████████████████████
| 15335/26400 [00:53<00:41, 266.17it/s]
 58%|██████████████████████████████████████████▍
| 15362/26400 [00:53<00:42, 260.57it/s]
 58%|██████████████████████████████████████████▍
| 15392/26400 [00:53<00:40, 268.55it/s]
 58%|██████████████████████████████████████████▌
| 15422/26400 [00:53<00:40, 272.94it/s]
 59%|██████████████████████████████████████████▋
| 15450/26400 [00:53<00:41, 261.38it/s]
 59%|██████████████████████████████████████████▋
| 15477/26400 [00:53<00:41, 263.37it/s]
 59%|███████████████████████████████████████████
| 15504/26400 [00:53<00:42, 257.21it/s]
 59%|███████████████████████████████████████████▏
| 15534/26400 [00:53<00:40, 268.18it/s]
 59%|███████████████████████████████████████████▏
| 15562/26400 [00:54<00:40, 267.17it/s]
 59%|███████████████████████████████████████████▎
| 15590/26400 [00:54<00:39, 270.32it/s]
 59%|███████████████████████████████████████████▎
| 15619/26400 [00:54<00:39, 274.58it/s]
 59%|███████████████████████████████████████████▍
| 15650/26400 [00:54<00:37, 283.76it/s]
 59%|███████████████████████████████████████████▌
| 15679/26400 [00:54<00:37, 282.50it/s]
```

```
60%|██████████████████████████████████████████████
| 15708/26400 [00:54<00:37, 284.10it/s]
60%|██████████████████████████████████████████████
| 15739/26400 [00:54<00:36, 289.18it/s]
60%|██████████████████████████████████████████████
| 15773/26400 [00:54<00:35, 301.37it/s]
60%|██████████████████████████████████████████████
| 15804/26400 [00:54<00:37, 284.13it/s]
60%|██████████████████████████████████████████████
| 15836/26400 [00:55<00:36, 292.62it/s]
60%|██████████████████████████████████████████████
| 15870/26400 [00:55<00:34, 304.79it/s]
60%|██████████████████████████████████████████████
| 15901/26400 [00:55<00:34, 301.23it/s]
60%|██████████████████████████████████████████████
| 15932/26400 [00:55<00:34, 302.27it/s]
60%|██████████████████████████████████████████████
| 15964/26400 [00:55<00:34, 306.73it/s]
61%|██████████████████████████████████████████████
| 15995/26400 [00:55<00:35, 293.97it/s]
61%|██████████████████████████████████████████████
| 16025/26400 [00:55<00:35, 293.39it/s]
61%|██████████████████████████████████████████████
| 16055/26400 [00:55<00:36, 286.29it/s]
61%|██████████████████████████████████████████████
| 16088/26400 [00:55<00:34, 295.95it/s]
61%|██████████████████████████████████████████████
| 16118/26400 [00:55<00:36, 284.72it/s]
61%|██████████████████████████████████████████████
| 16147/26400 [00:56<00:36, 280.70it/s]
61%|██████████████████████████████████████████████
| 16178/26400 [00:56<00:35, 285.92it/s]
61%|██████████████████████████████████████████████
| 16213/26400 [00:56<00:33, 301.19it/s]
62%|██████████████████████████████████████████████
| 16244/26400 [00:56<00:33, 303.13it/s]
62%|██████████████████████████████████████████████
| 16275/26400 [00:56<00:34, 291.64it/s]
62%|██████████████████████████████████████████████
| 16308/26400 [00:56<00:33, 301.57it/s]
62%|██████████████████████████████████████████████
| 16339/26400 [00:56<00:33, 297.31it/s]
62%|██████████████████████████████████████████████
| 16370/26400 [00:56<00:33, 298.63it/s]
62%|██████████████████████████████████████████████
| 16401/26400 [00:56<00:33, 295.30it/s]
62%|██████████████████████████████████████████████
| 16432/26400 [00:57<00:33, 298.93it/s]
62%|██████████████████████████████████████████████
| 16462/26400 [00:57<00:35, 277.14it/s]
62%|██████████████████████████████████████████████
| 16491/26400 [00:57<00:36, 273.17it/s]
63%|██████████████████████████████████████████████
| 16519/26400 [00:57<00:36, 272.99it/s]
63%|██████████████████████████████████████████████
| 16552/26400 [00:57<00:34, 281.50it/s]
63%|██████████████████████████████████████████████
```

```
| 16582/26400 [00:57<00:34, 286.20it/s]
 63%|████████████████████████████████████████████████▏
| 16611/26400 [00:57<00:34, 280.08it/s]
 63%|████████████████████████████████████████████████▎
| 16640/26400 [00:57<00:36, 269.80it/s]
 63%|████████████████████████████████████████████████▎
| 16673/26400 [00:57<00:34, 284.88it/s]
 63%|████████████████████████████████████████████████▍
| 16702/26400 [00:58<00:34, 284.94it/s]
 63%|█████████████████████████████████████████████████
| 16736/26400 [00:58<00:32, 298.91it/s]
 64%|████████████████████████████████████████████████▋
| 16767/26400 [00:58<00:33, 288.90it/s]
 64%|████████████████████████████████████████████████▋
| 16797/26400 [00:58<00:33, 285.71it/s]
 64%|████████████████████████████████████████████████▊
| 16826/26400 [00:58<00:33, 284.68it/s]
 64%|████████████████████████████████████████████████▉
| 16855/26400 [00:58<00:33, 284.80it/s]
 64%|████████████████████████████████████████████████▉
| 16887/26400 [00:58<00:32, 293.12it/s]
 64%|██████████████████████████████████████████████████
| 16921/26400 [00:58<00:31, 304.34it/s]
 64%|█████████████████████████████████████████████████▏
| 16952/26400 [00:58<00:31, 304.47it/s]
 64%|█████████████████████████████████████████████████▏
| 16983/26400 [00:58<00:32, 292.51it/s]
 64%|█████████████████████████████████████████████████▎
| 17013/26400 [00:59<00:32, 285.70it/s]
 65%|█████████████████████████████████████████████████▍
| 17042/26400 [00:59<00:32, 285.52it/s]
 65%|█████████████████████████████████████████████████▍
| 17071/26400 [00:59<00:35, 262.93it/s]
 65%|██████████████████████████████████████████████████
| 17098/26400 [00:59<00:35, 262.90it/s]
 65%|█████████████████████████████████████████████████▋
| 17127/26400 [00:59<00:34, 269.93it/s]
 65%|█████████████████████████████████████████████████▋
| 17157/26400 [00:59<00:33, 277.74it/s]
 65%|█████████████████████████████████████████████████▊
| 17190/26400 [00:59<00:31, 287.99it/s]
 65%|█████████████████████████████████████████████████▉
| 17221/26400 [00:59<00:31, 291.99it/s]
 65%|███████████████████████████████████████████████████
| 17251/26400 [00:59<00:32, 282.11it/s]
 65%|███████████████████████████████████████████████████
| 17282/26400 [01:00<00:31, 289.36it/s]
 66%|██████████████████████████████████████████████████▏
| 17312/26400 [01:00<00:32, 283.61it/s]
 66%|██████████████████████████████████████████████████▎
| 17343/26400 [01:00<00:31, 289.63it/s]
 66%|██████████████████████████████████████████████████▎
| 17373/26400 [01:00<00:31, 288.66it/s]
 66%|██████████████████████████████████████████████████▍
| 17405/26400 [01:00<00:30, 295.99it/s]
 66%|████████████████████████████████████████████████████
| 17435/26400 [01:00<00:30, 293.06it/s]
```

```
66%|
| 17465/26400 [01:00<00:31, 285.27it/s]
66%|
| 17496/26400 [01:00<00:31, 286.82it/s]
66%|
| 17526/26400 [01:00<00:30, 288.36it/s]
67%|
| 17558/26400 [01:00<00:29, 295.76it/s]
67%|
| 17588/26400 [01:01<00:30, 284.60it/s]
67%|
| 17620/26400 [01:01<00:30, 291.37it/s]
67%|
| 17650/26400 [01:01<00:30, 289.05it/s]
67%|
| 17680/26400 [01:01<00:29, 291.63it/s]
67%|
| 17710/26400 [01:01<00:30, 282.70it/s]
67%|
| 17739/26400 [01:01<00:31, 279.32it/s]
67%|
| 17771/26400 [01:01<00:29, 289.83it/s]
67%|
| 17801/26400 [01:01<00:30, 286.34it/s]
68%|
| 17830/26400 [01:01<00:31, 275.40it/s]
68%|
| 17858/26400 [01:02<00:31, 270.57it/s]
68%|
| 17886/26400 [01:02<00:31, 268.06it/s]
68%|
| 17919/26400 [01:02<00:29, 282.80it/s]
68%|
| 17954/26400 [01:02<00:28, 297.23it/s]
68%|
| 17985/26400 [01:02<00:28, 300.32it/s]
68%|
| 18016/26400 [01:02<00:28, 292.27it/s]
68%|
| 18046/26400 [01:02<00:29, 285.54it/s]
68%|
| 18076/26400 [01:02<00:28, 288.25it/s]
69%|
| 18105/26400 [01:02<00:28, 286.48it/s]
69%|
| 18137/26400 [01:02<00:27, 295.18it/s]
69%|
| 18167/26400 [01:03<00:28, 291.66it/s]
69%|
| 18197/26400 [01:03<00:29, 280.35it/s]
69%|
| 18226/26400 [01:03<00:29, 274.57it/s]
69%|
| 18257/26400 [01:03<00:28, 283.75it/s]
69%|
| 18289/26400 [01:03<00:27, 291.54it/s]
69%|
```

```
| 18322/26400 [01:03<00:27, 297.43it/s]
 70%|
| 18353/26400 [01:03<00:26, 299.58it/s]
 70%|
| 18384/26400 [01:03<00:27, 290.97it/s]
 70%|
| 18414/26400 [01:03<00:27, 290.43it/s]
 70%|
| 18445/26400 [01:04<00:27, 288.04it/s]
 70%|
| 18476/26400 [01:04<00:26, 293.68it/s]
 70%|
| 18507/26400 [01:04<00:27, 291.89it/s]
 70%|
| 18537/26400 [01:04<00:27, 285.28it/s]
 70%|
| 18566/26400 [01:04<00:27, 280.28it/s]
 70%|
| 18597/26400 [01:04<00:27, 287.99it/s]
 71%|
| 18629/26400 [01:04<00:26, 289.09it/s]
 71%|
| 18658/26400 [01:04<00:27, 286.17it/s]
 71%|
| 18690/26400 [01:04<00:26, 293.33it/s]
 71%|
| 18723/26400 [01:05<00:25, 301.18it/s]
 71%|
| 18756/26400 [01:05<00:24, 307.79it/s]
 71%|
| 18790/26400 [01:05<00:24, 314.40it/s]
 71%|
| 18822/26400 [01:05<00:25, 298.64it/s]
 71%|
| 18853/26400 [01:05<00:25, 291.16it/s]
 72%|
| 18884/26400 [01:05<00:25, 295.10it/s]
 72%|
| 18914/26400 [01:05<00:27, 274.07it/s]
 72%|
| 18947/26400 [01:05<00:25, 287.45it/s]
 72%|
| 18977/26400 [01:05<00:26, 277.61it/s]
 72%|
| 19006/26400 [01:05<00:26, 279.01it/s]
 72%|
| 19039/26400 [01:06<00:25, 288.95it/s]
 72%|
| 19070/26400 [01:06<00:25, 291.85it/s]
 72%|
| 19102/26400 [01:06<00:24, 297.49it/s]
 72%|
| 19135/26400 [01:06<00:24, 300.91it/s]
 73%|
| 19166/26400 [01:06<00:25, 288.59it/s]
 73%|
| 19196/26400 [01:06<00:25, 285.50it/s]
```

```
73%|████████████████████████████████████████▉
| 19229/26400 [01:06<00:24, 295.34it/s]
73%|████████████████████████████████████████▉
| 19259/26400 [01:06<00:24, 291.79it/s]
73%|████████████████████████████████████████▉
| 19291/26400 [01:06<00:24, 293.36it/s]
73%|█████████████████████████████████████████
| 19321/26400 [01:07<00:26, 263.00it/s]
73%|█████████████████████████████████████████
| 19349/26400 [01:07<00:26, 266.55it/s]
73%|█████████████████████████████████████████
| 19377/26400 [01:07<00:27, 256.56it/s]
74%|█████████████████████████████████████████▏
| 19404/26400 [01:07<00:27, 252.62it/s]
74%|█████████████████████████████████████████▏
| 19435/26400 [01:07<00:26, 264.92it/s]
74%|█████████████████████████████████████████▎
| 19471/26400 [01:07<00:24, 286.51it/s]
74%|█████████████████████████████████████████▍
| 19504/26400 [01:07<00:23, 297.74it/s]
74%|█████████████████████████████████████████▍
| 19535/26400 [01:07<00:23, 295.53it/s]
74%|█████████████████████████████████████████▍
| 19566/26400 [01:07<00:23, 294.84it/s]
74%|█████████████████████████████████████████▋
| 19596/26400 [01:08<00:23, 294.84it/s]
74%|█████████████████████████████████████████▊
| 19626/26400 [01:08<00:23, 289.76it/s]
74%|█████████████████████████████████████████▊
| 19656/26400 [01:08<00:24, 279.90it/s]
75%|█████████████████████████████████████████▉
| 19687/26400 [01:08<00:23, 283.00it/s]
75%|██████████████████████████████████████████
| 19723/26400 [01:08<00:22, 299.59it/s]
75%|██████████████████████████████████████████
| 19754/26400 [01:08<00:22, 300.24it/s]
75%|██████████████████████████████████████████▏
| 19785/26400 [01:08<00:22, 292.22it/s]
75%|██████████████████████████████████████████▎
| 19815/26400 [01:08<00:23, 283.89it/s]
75%|██████████████████████████████████████████▍
| 19844/26400 [01:08<00:23, 283.40it/s]
75%|██████████████████████████████████████████▍
| 19873/26400 [01:09<00:22, 284.75it/s]
75%|██████████████████████████████████████████▍
| 19909/26400 [01:09<00:21, 301.72it/s]
76%|██████████████████████████████████████████▊
| 19940/26400 [01:09<00:21, 303.50it/s]
76%|██████████████████████████████████████████▊
| 19971/26400 [01:09<00:22, 283.89it/s]
76%|██████████████████████████████████████████▉
| 20004/26400 [01:09<00:21, 292.59it/s]
76%|███████████████████████████████████████████
| 20034/26400 [01:09<00:21, 291.57it/s]
76%|███████████████████████████████████████████
| 20064/26400 [01:09<00:22, 282.67it/s]
76%|███████████████████████████████████████████
```

```
| 20093/26400 [01:09<00:22, 283.39it/s]
76%|
| 20123/26400 [01:09<00:21, 286.75it/s]
76%|
| 20152/26400 [01:09<00:23, 271.03it/s]
76%|
| 20180/26400 [01:10<00:23, 269.15it/s]
77%|
| 20209/26400 [01:10<00:22, 274.51it/s]
77%|
| 20241/26400 [01:10<00:21, 284.66it/s]
77%|
| 20272/26400 [01:10<00:21, 291.21it/s]
77%|
| 20302/26400 [01:10<00:21, 283.23it/s]
77%|
| 20331/26400 [01:10<00:21, 278.88it/s]
77%|
| 20360/26400 [01:10<00:22, 266.80it/s]
77%|
| 20392/26400 [01:10<00:21, 277.36it/s]
77%|
| 20423/26400 [01:10<00:20, 285.83it/s]
77%|
| 20458/26400 [01:11<00:19, 301.89it/s]
78%|
| 20491/26400 [01:11<00:19, 309.17it/s]
78%|
| 20525/26400 [01:11<00:18, 317.17it/s]
78%|
| 20558/26400 [01:11<00:19, 292.20it/s]
78%|
| 20588/26400 [01:11<00:19, 292.18it/s]
78%|
| 20623/26400 [01:11<00:18, 306.03it/s]
78%|
| 20655/26400 [01:11<00:19, 289.34it/s]
78%|
| 20685/26400 [01:11<00:20, 285.19it/s]
78%|
| 20718/26400 [01:11<00:19, 294.34it/s]
79%|
| 20748/26400 [01:12<00:19, 289.41it/s]
79%|
| 20778/26400 [01:12<00:19, 287.70it/s]
79%|
| 20807/26400 [01:12<00:19, 282.72it/s]
79%|
| 20838/26400 [01:12<00:19, 288.18it/s]
79%|
| 20868/26400 [01:12<00:19, 288.50it/s]
79%|
| 20900/26400 [01:12<00:18, 294.22it/s]
79%|
| 20930/26400 [01:12<00:19, 277.33it/s]
79%|
| 20961/26400 [01:12<00:19, 285.80it/s]
```

```
 80%|
| 20990/26400 [01:12<00:18, 286.43it/s]
 80%|
| 21024/26400 [01:12<00:18, 298.48it/s]
 80%|
| 21055/26400 [01:13<00:18, 296.04it/s]
 80%|
| 21086/26400 [01:13<00:17, 298.60it/s]
 80%|
| 21119/26400 [01:13<00:17, 306.74it/s]
 80%|
| 21151/26400 [01:13<00:17, 308.16it/s]
 80%|
| 21182/26400 [01:13<00:17, 301.76it/s]
 80%|
| 21213/26400 [01:13<00:17, 297.43it/s]
 80%|
| 21244/26400 [01:13<00:17, 300.45it/s]
 81%|
| 21275/26400 [01:13<00:17, 289.10it/s]
 81%|
| 21309/26400 [01:13<00:16, 301.30it/s]
 81%|
| 21342/26400 [01:14<00:16, 308.71it/s]
 81%|
| 21374/26400 [01:14<00:16, 306.03it/s]
 81%|
| 21405/26400 [01:14<00:17, 286.22it/s]
 81%|
| 21441/26400 [01:14<00:16, 302.87it/s]
 81%|
| 21472/26400 [01:14<00:16, 293.95it/s]
 81%|
| 21504/26400 [01:14<00:16, 299.86it/s]
 82%|
| 21535/26400 [01:14<00:16, 294.45it/s]
 82%|
| 21565/26400 [01:14<00:16, 286.18it/s]
 82%|
| 21596/26400 [01:14<00:16, 292.33it/s]
 82%|
| 21627/26400 [01:14<00:16, 296.79it/s]
 82%|
| 21657/26400 [01:15<00:16, 296.22it/s]
 82%|
| 21687/26400 [01:15<00:16, 291.52it/s]
 82%|
| 21720/26400 [01:15<00:15, 300.66it/s]
 82%|
| 21751/26400 [01:15<00:15, 290.86it/s]
 83%|
| 21781/26400 [01:15<00:16, 281.40it/s]
 83%|
| 21810/26400 [01:15<00:16, 272.95it/s]
 83%|
| 21839/26400 [01:15<00:16, 276.48it/s]
 83%|
```

```
| 21867/26400 [01:15<00:16, 275.29it/s]
83%|████████▎
| 21895/26400 [01:15<00:16, 273.67it/s]
83%|████████▎
| 21927/26400 [01:16<00:15, 282.52it/s]
83%|████████▎
| 21956/26400 [01:16<00:15, 279.20it/s]
83%|████████▎
| 21985/26400 [01:16<00:16, 271.49it/s]
83%|████████▎
| 22013/26400 [01:16<00:16, 271.82it/s]
84%|████████▎
| 22047/26400 [01:16<00:15, 286.49it/s]
84%|████████▎
| 22076/26400 [01:16<00:15, 277.87it/s]
84%|████████▎
| 22105/26400 [01:16<00:15, 279.99it/s]
84%|████████▍
| 22138/26400 [01:16<00:14, 292.75it/s]
84%|████████▍
| 22168/26400 [01:16<00:14, 294.26it/s]
84%|████████▍
| 22198/26400 [01:17<00:14, 287.69it/s]
84%|████████▍
| 22227/26400 [01:17<00:14, 281.90it/s]
84%|████████▍
| 22256/26400 [01:17<00:14, 281.20it/s]
84%|████████▍
| 22287/26400 [01:17<00:14, 287.87it/s]
85%|████████▍
| 22316/26400 [01:17<00:14, 287.03it/s]
85%|████████▍
| 22345/26400 [01:17<00:14, 274.28it/s]
85%|████████▍
| 22377/26400 [01:17<00:14, 285.24it/s]
85%|████████▍
| 22409/26400 [01:17<00:13, 293.45it/s]
85%|████████▍
| 22439/26400 [01:17<00:13, 293.88it/s]
85%|████████▍
| 22469/26400 [01:17<00:14, 273.34it/s]
85%|████████▌
| 22497/26400 [01:18<00:14, 270.74it/s]
85%|████████▌
| 22528/26400 [01:18<00:13, 279.36it/s]
85%|████████▌
| 22557/26400 [01:18<00:13, 277.83it/s]
86%|████████▌
| 22585/26400 [01:18<00:14, 266.02it/s]
86%|████████▌
| 22620/26400 [01:18<00:13, 285.44it/s]
86%|████████▌
| 22650/26400 [01:18<00:14, 266.03it/s]
86%|████████▌
| 22681/26400 [01:18<00:13, 275.84it/s]
86%|████████▌
| 22712/26400 [01:18<00:12, 283.92it/s]
```

```
86%|█████████
| 22745/26400 [01:18<00:12, 294.17it/s]
86%|█████████
| 22775/26400 [01:19<00:12, 291.82it/s]
86%|█████████
| 22806/26400 [01:19<00:12, 293.90it/s]
86%|█████████
| 22836/26400 [01:19<00:12, 289.11it/s]
87%|█████████
| 22868/26400 [01:19<00:11, 296.31it/s]
87%|█████████
| 22898/26400 [01:19<00:11, 292.43it/s]
87%|█████████
| 22928/26400 [01:19<00:12, 280.85it/s]
87%|█████████
| 22958/26400 [01:19<00:12, 285.74it/s]
87%|█████████
| 22987/26400 [01:19<00:12, 276.58it/s]
87%|█████████
| 23017/26400 [01:19<00:11, 282.60it/s]
87%|█████████
| 23046/26400 [01:20<00:11, 280.09it/s]
87%|█████████
| 23076/26400 [01:20<00:11, 284.37it/s]
88%|█████████
| 23107/26400 [01:20<00:11, 287.77it/s]
88%|█████████
| 23136/26400 [01:20<00:11, 281.95it/s]
88%|█████████
| 23165/26400 [01:20<00:11, 281.24it/s]
88%|█████████
| 23194/26400 [01:20<00:12, 261.08it/s]
88%|█████████
| 23226/26400 [01:20<00:11, 272.31it/s]
88%|█████████
| 23258/26400 [01:20<00:11, 283.75it/s]
88%|█████████
| 23287/26400 [01:20<00:11, 275.27it/s]
88%|█████████
| 23315/26400 [01:20<00:11, 265.10it/s]
88%|█████████
| 23347/26400 [01:21<00:10, 278.97it/s]
89%|█████████
| 23376/26400 [01:21<00:10, 277.56it/s]
89%|█████████
| 23405/26400 [01:21<00:10, 278.97it/s]
89%|█████████
| 23434/26400 [01:21<00:10, 279.96it/s]
89%|█████████
| 23463/26400 [01:21<00:10, 278.24it/s]
89%|█████████
| 23491/26400 [01:21<00:10, 273.29it/s]
89%|█████████
| 23522/26400 [01:21<00:10, 282.79it/s]
89%|█████████
| 23554/26400 [01:21<00:09, 292.42it/s]
89%|█████████
```

```
| 23585/26400 [01:21<00:09, 296.86it/s]
89%|
| 23620/26400 [01:22<00:09, 305.56it/s]
90%|
| 23651/26400 [01:22<00:08, 306.21it/s]
90%|
| 23682/26400 [01:22<00:09, 298.70it/s]
90%|
| 23712/26400 [01:22<00:09, 289.83it/s]
90%|
| 23747/26400 [01:22<00:08, 302.64it/s]
90%|
| 23778/26400 [01:22<00:08, 299.75it/s]
90%|
| 23812/26400 [01:22<00:08, 305.99it/s]
90%|
| 23843/26400 [01:22<00:08, 304.72it/s]
90%|
| 23874/26400 [01:22<00:08, 304.72it/s]
91%|
| 23905/26400 [01:22<00:08, 288.60it/s]
91%|
| 23936/26400 [01:23<00:08, 294.09it/s]
91%|
| 23966/26400 [01:23<00:08, 293.47it/s]
91%|
| 23997/26400 [01:23<00:08, 294.23it/s]
91%|
| 24027/26400 [01:23<00:08, 295.30it/s]
91%|
| 24057/26400 [01:23<00:08, 279.55it/s]
91%|
| 24091/26400 [01:23<00:07, 293.97it/s]
91%|
| 24121/26400 [01:23<00:08, 279.49it/s]
91%|
| 24152/26400 [01:23<00:07, 285.03it/s]
92%|
| 24181/26400 [01:23<00:07, 279.30it/s]
92%|
| 24210/26400 [01:24<00:07, 276.20it/s]
92%|
| 24238/26400 [01:24<00:07, 275.91it/s]
92%|
| 24266/26400 [01:24<00:08, 259.65it/s]
92%|
| 24293/26400 [01:24<00:08, 262.11it/s]
92%|
| 24325/26400 [01:24<00:07, 275.91it/s]
92%|
| 24358/26400 [01:24<00:07, 282.94it/s]
92%|
| 24389/26400 [01:24<00:06, 288.34it/s]
92%|
| 24419/26400 [01:24<00:07, 269.96it/s]
93%|
| 24447/26400 [01:24<00:07, 272.31it/s]
```

```
93%|█████████▎
| 24475/26400 [01:25<00:07, 270.81it/s]
93%|█████████▎
| 24508/26400 [01:25<00:06, 284.20it/s]
93%|█████████▎
| 24543/26400 [01:25<00:06, 299.08it/s]
93%|█████████▎
| 24574/26400 [01:25<00:06, 295.60it/s]
93%|█████████▎
| 24604/26400 [01:25<00:06, 285.31it/s]
93%|█████████▎
| 24634/26400 [01:25<00:06, 288.95it/s]
93%|█████████▎
| 24664/26400 [01:25<00:05, 291.56it/s]
94%|█████████▎
| 24698/26400 [01:25<00:05, 303.98it/s]
94%|█████████▎
| 24729/26400 [01:25<00:05, 293.02it/s]
94%|█████████▎
| 24759/26400 [01:25<00:05, 282.02it/s]
94%|█████████▍
| 24789/26400 [01:26<00:05, 286.58it/s]
94%|█████████▍
| 24818/26400 [01:26<00:05, 282.79it/s]
94%|█████████▍
| 24847/26400 [01:26<00:05, 274.62it/s]
94%|█████████▍
| 24875/26400 [01:26<00:05, 275.63it/s]
94%|█████████▍
| 24903/26400 [01:26<00:05, 257.34it/s]
94%|█████████▍
| 24930/26400 [01:26<00:05, 258.96it/s]
95%|█████████▍
| 24965/26400 [01:26<00:05, 277.72it/s]
95%|█████████▍
| 24994/26400 [01:26<00:05, 279.89it/s]
95%|█████████▍
| 25024/26400 [01:26<00:04, 284.22it/s]
95%|█████████▍
| 25053/26400 [01:27<00:04, 277.96it/s]
95%|█████████▌
| 25087/26400 [01:27<00:04, 292.74it/s]
95%|█████████▌
| 25117/26400 [01:27<00:04, 291.68it/s]
95%|█████████▌
| 25148/26400 [01:27<00:04, 293.80it/s]
95%|█████████▌
| 25178/26400 [01:27<00:04, 289.04it/s]
95%|█████████▌
| 25208/26400 [01:27<00:04, 287.44it/s]
96%|█████████▌
| 25237/26400 [01:27<00:04, 285.04it/s]
96%|█████████▌
| 25269/26400 [01:27<00:03, 294.11it/s]
96%|█████████▌
| 25299/26400 [01:27<00:03, 287.59it/s]
96%|█████████▌
```

```
| 25328/26400 [01:28<00:03, 280.20it/s]
 96%|████████████████████████████████████████████████████████████████████████
| 25357/26400 [01:28<00:03, 273.69it/s]
 96%|████████████████████████████████████████████████████████████████████████
▏   | 25388/26400 [01:28<00:03, 283.09it/s]
 96%|████████████████████████████████████████████████████████████████████████
▏   | 25417/26400 [01:28<00:03, 284.51it/s]
 96%|████████████████████████████████████████████████████████████████████████
▍   | 25446/26400 [01:28<00:03, 264.48it/s]
 97%|████████████████████████████████████████████████████████████████████████
▌   | 25477/26400 [01:28<00:03, 276.13it/s]
 97%|████████████████████████████████████████████████████████████████████████
▌   | 25508/26400 [01:28<00:03, 283.36it/s]
 97%|████████████████████████████████████████████████████████████████████████
▌  | 25538/26400 [01:28<00:03, 286.72it/s]
 97%|████████████████████████████████████████████████████████████████████████
█   | 25567/26400 [01:28<00:02, 285.38it/s]
 97%|████████████████████████████████████████████████████████████████████████
█   | 25596/26400 [01:28<00:02, 279.53it/s]
 97%|████████████████████████████████████████████████████████████████████████
█   | 25627/26400 [01:29<00:02, 287.44it/s]
 97%|████████████████████████████████████████████████████████████████████████
█▌  | 25656/26400 [01:29<00:02, 285.88it/s]
 97%|████████████████████████████████████████████████████████████████████████
█▌  | 25686/26400 [01:29<00:02, 287.70it/s]
 97%|████████████████████████████████████████████████████████████████████████
█  | 25715/26400 [01:29<00:02, 284.38it/s]
 98%|████████████████████████████████████████████████████████████████████████
█   | 25745/26400 [01:29<00:02, 288.29it/s]
 98%|████████████████████████████████████████████████████████████████████████
█   | 25774/26400 [01:29<00:02, 275.87it/s]
 98%|████████████████████████████████████████████████████████████████████████
█▍  | 25804/26400 [01:29<00:02, 279.74it/s]
 98%|████████████████████████████████████████████████████████████████████████
█▋  | 25833/26400 [01:29<00:02, 278.09it/s]
 98%|████████████████████████████████████████████████████████████████████████
█▋  | 25861/26400 [01:29<00:01, 276.41it/s]
 98%|████████████████████████████████████████████████████████████████████████
█▌ | 25891/26400 [01:30<00:01, 281.71it/s]
 98%|████████████████████████████████████████████████████████████████████████
██  | 25924/26400 [01:30<00:01, 292.51it/s]
 98%|████████████████████████████████████████████████████████████████████████
██  | 25956/26400 [01:30<00:01, 299.63it/s]
 98%|████████████████████████████████████████████████████████████████████████
██▍ | 25987/26400 [01:30<00:01, 292.64it/s]
 99%|████████████████████████████████████████████████████████████████████████
██▌ | 26017/26400 [01:30<00:01, 294.18it/s]
 99%|████████████████████████████████████████████████████████████████████████
██▌ | 26047/26400 [01:30<00:01, 288.47it/s]
 99%|████████████████████████████████████████████████████████████████████████
██ | 26076/26400 [01:30<00:01, 279.97it/s]
 99%|████████████████████████████████████████████████████████████████████████
██▏ | 26105/26400 [01:30<00:01, 271.24it/s]
 99%|████████████████████████████████████████████████████████████████████████
██▍ | 26138/26400 [01:30<00:00, 283.77it/s]
 99%|████████████████████████████████████████████████████████████████████████
██▍ | 26170/26400 [01:30<00:00, 292.38it/s]
```

```
 99%|██████████████████████████████████████████████████████████████████████
███ | 26200/26400 [01:31<00:00, 288.91it/s]
 99%|██████████████████████████████████████████████████████████████████████
███| 26230/26400 [01:31<00:00, 284.88it/s]
 99%|██████████████████████████████████████████████████████████████████████
███| 26265/26400 [01:31<00:00, 299.63it/s]
100%|██████████████████████████████████████████████████████████████████████
███ | 26296/26400 [01:31<00:00, 301.14it/s]
100%|██████████████████████████████████████████████████████████████████████
███ | 26330/26400 [01:31<00:00, 311.21it/s]
100%|██████████████████████████████████████████████████████████████████████
████| 26362/26400 [01:31<00:00, 303.35it/s]
100%|██████████████████████████████████████████████████████████████████████
███| 26400/26400 [01:31<00:00, 287.86it/s]

26400
300
```

In [119]:

```
# preprocessing for Project_title with TFIDF Vectorization
tfidf_model = TfidfVectorizer()
tfidf_model.fit(X_train['project_title'])
# we are converting a dictionary with word as a key, and the idf as a value
dictionary = dict(zip(tfidf_model.get_feature_names(), list(tfidf_model.idf_
)))
tfidf_words = set(tfidf_model.get_feature_names())
```

In [120]:
```
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_pj_title_train = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_train['project_title']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_pj_title_train.append(vector)

print(len(tfidf_w2v_vectors_pj_title_train))
print(len(tfidf_w2v_vectors_pj_title_train[0]))
```

```
  0%|
| 0/35912 [00:00<?, ?it/s]
 28%|████████████████████▌
| 10034/35912 [00:00<00:00, 99611.70it/s]
 55%|████████████████████████████████████████
| 19722/35912 [00:00<00:00, 98555.89it/s]
100%|████████████████████████████████████████████████████████████████████████
█| 35912/35912 [00:00<00:00, 97318.98it/s]

35912
300
```

In [121]:
```python
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_pj_title_cv = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_cv['project_title']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_pj_title_cv.append(vector)

print(len(tfidf_w2v_vectors_pj_title_cv))
print(len(tfidf_w2v_vectors_pj_title_cv[0]))
```

```
  0%|
| 0/17688 [00:00<?, ?it/s]
100%|████████████████████████████████████████████████████████████████████
█| 17688/17688 [00:00<00:00, 94340.56it/s]

17688
300
```

In [122]:

```python
# average Word2Vec
# compute average word2vec for each review.
tfidf_w2v_vectors_pj_title_test = []; # the avg-w2v for each sentence/review is stored in this list
for sentence in tqdm(X_test['project_title']): # for each review/sentence
    vector = np.zeros(300) # as word vectors are of zero length
    tf_idf_weight =0; # num of words with a valid vector in the sentence/review
    for word in sentence.split(): # for each word in a review/sentence
        if (word in glove_words) and (word in tfidf_words):
            vec = model[word] # getting the vector for each word
            # here we are multiplying idf value(dictionary[word]) and the tf value((sentence.count(word)/len(sentence.split())))
            tf_idf = dictionary[word]*(sentence.count(word)/len(sentence.split())) # getting the tfidf value for each word
            vector += (vec * tf_idf) # calculating tfidf weighted w2v
            tf_idf_weight += tf_idf
    if tf_idf_weight != 0:
        vector /= tf_idf_weight
    tfidf_w2v_vectors_pj_title_test.append(vector)

print(len(tfidf_w2v_vectors_pj_title_test))
print(len(tfidf_w2v_vectors_pj_title_test[0]))
```

```
  0%|
| 0/26400 [00:00<?, ?it/s]
 32%|████████████████████████
| 8398/26400 [00:00<00:00, 83384.07it/s]
100%|██████████████████████████████████████████████████████████████████████████
█| 26400/26400 [00:00<00:00, 93872.79it/s]

26400
300
```

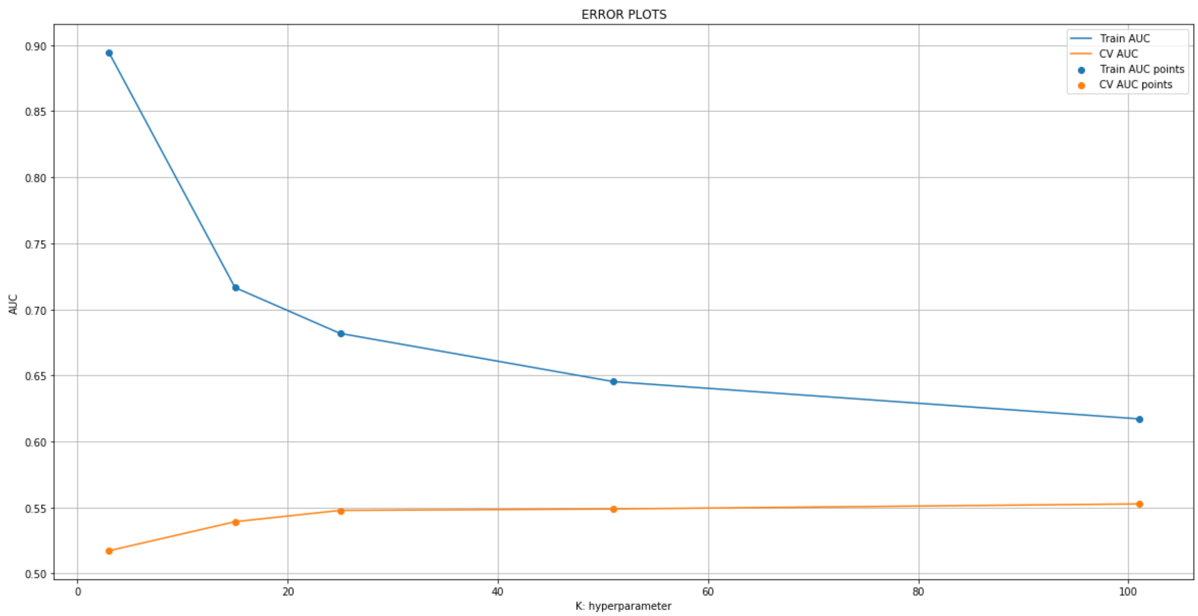
In [123]:
```
# Concatinating all the features
X_tr = hstack((tfidf_w2v_vectors_train, X_train_state_ohe, X_train_teacher_ohe
,
               X_train_grade_ohe, X_train_category_ohe,
               X_train_subcategory_ohe, X_train_price_norm,
               X_train_teach_prev_norm, tfidf_w2v_vectors_pj_title_train)).toc
sr()

X_cr = hstack((tfidf_w2v_vectors_cv, X_cv_state_ohe, X_cv_teacher_ohe,
               X_cv_grade_ohe,X_cv_category_ohe,
               X_cv_subcategory_ohe, X_cv_price_norm,
              X_cv_teach_prev_norm, tfidf_w2v_vectors_pj_title_cv)).tocsr()

X_te = hstack((tfidf_w2v_vectors_test, X_test_state_ohe, X_test_teacher_ohe,
               X_test_grade_ohe, X_test_category_ohe,
               X_test_subcategory_ohe, X_test_price_norm,
              X_test_teach_prev_norm, tfidf_w2v_vectors_pj_title_test)).tocsr
()

print("Final Data matrix")
print(X_tr.shape, y_train.shape)
print(X_cr.shape, y_cv.shape)
print(X_te.shape, y_test.shape)
print("="*100)
```

```
Final Data matrix
(35912, 701) (35912,)
(17688, 701) (17688,)
(26400, 701) (26400,)
============================================================================
=======================
```

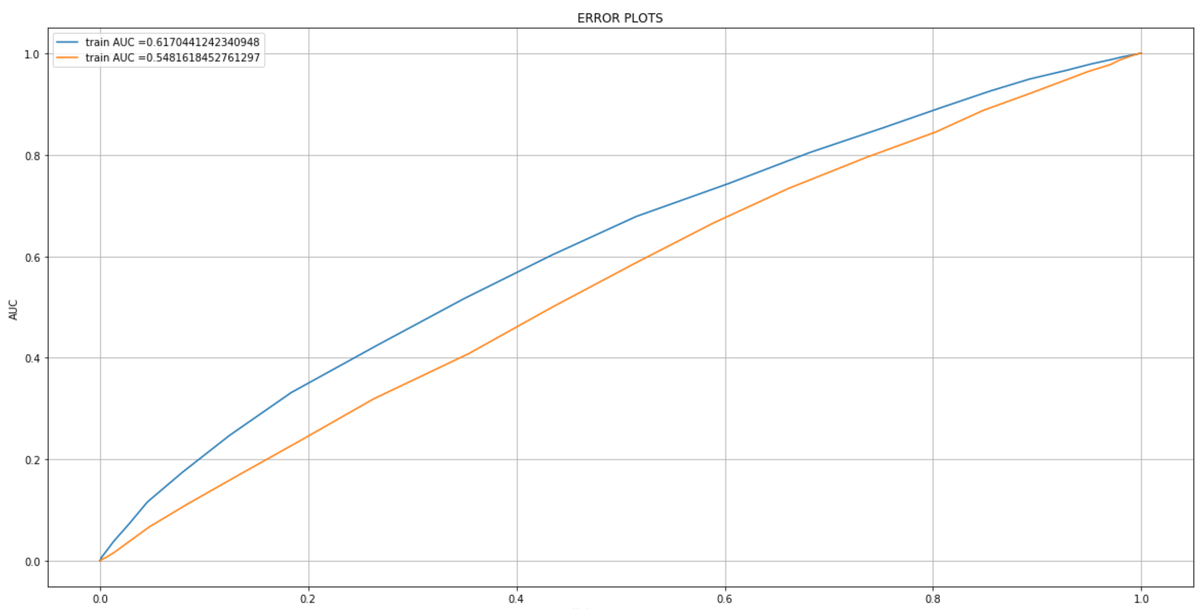
In [124]:
```python
# Hyper parameter Tuning
train_auc = []
cv_auc = []
K = [3, 15, 25, 51, 101]
for i in tqdm(K):
    neigh = KNeighborsClassifier(n_neighbors=i, n_jobs=-1)
    neigh.fit(X_tr, y_train)

    y_train_pred = batch_predict(neigh, X_tr)
    y_cv_pred = batch_predict(neigh, X_cr)

    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
 20%|
| 1/5 [07:01<28:04, 421.02s/it]
 40%|
| 2/5 [14:10<21:10, 423.59s/it]
 60%|
| 3/5 [21:23<14:12, 426.24s/it]
 80%|
| 4/5 [28:34<07:07, 427.76s/it]
100%|
| 5/5 [35:44<00:00, 428.98s/it]
```

In [125]:
```
plt.figure(figsize=(20,10))
plt.plot(K, train_auc, label='Train AUC')
plt.plot(K, cv_auc, label='CV AUC')

plt.scatter(K, train_auc, label='Train AUC points')
plt.scatter(K, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



In [126]:
```
best_k = 101
from sklearn.metrics import roc_curve, auc


neigh = KNeighborsClassifier(n_neighbors=best_k, n_jobs=-1)
neigh.fit(X_tr, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estim
ates of the positive class
# not the predicted outputs

y_train_pred = batch_predict(neigh, X_tr)
y_test_pred = batch_predict(neigh, X_te)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
plt.figure(figsize=(20,10))
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tp
r)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



The Blue plot in the Graph is Train AUC and Orange Plot is Test AUC. Initially the label was accidentally copy pasted as Train AUC for both. Since redrawing plot would require to re run entire ipynb and it takes up a lot of time. so documenting the information as a markdown.

2.5 Feature selection with `SelectKBest`

In [127]:
```
# please write all the code with proper documentation, and proper titles for e
ach subsection
# go through documentations and blogs before you start coding
# first figure out what to do, and then think about how to do.
# reading and understanding error messages will be very much helpfull in debug
ging your code

# when you plot any graph make sure you use
    # a. Title, that describes your plot, this will be very helpful to the rea
der
    # b. Legends if needed
    # c. X-axis label
    # d. Y-axis label

# Performing the Concatination of Features for Set-2

X_tr = hstack((X_train_essay_tfidf, X_train_state_ohe, X_train_teacher_ohe,
               X_train_grade_ohe, X_train_price_norm, X_train_category_ohe,
               X_train_subcategory_ohe, X_train_teach_prev_norm,
               X_train_pj_title_tfidf)).tocsr()

X_cr = hstack((X_cv_essay_tfidf, X_cv_state_ohe, X_cv_teacher_ohe,
               X_cv_grade_ohe, X_cv_category_ohe, X_cv_subcategory_ohe,
               X_cv_price_norm, X_cv_teach_prev_norm, X_cv_pj_title_tfidf)).to
csr()

X_te = hstack((X_test_essay_tfidf, X_test_state_ohe, X_test_teacher_ohe,
               X_test_grade_ohe, X_test_category_ohe, X_test_subcategory_ohe,
               X_test_price_norm, X_test_teach_prev_norm,
               X_test_pj_title_tfidf)).tocsr()
```

In [128]:
```
# importing necessary packages for Univariate feature selection
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
```

In [129]:
```
print ("shape before selecting top features")
print (X_tr.shape)
print (X_cr.shape)
print (X_te.shape)
```

```
shape before selecting top features
(35912, 10101)
(17688, 10101)
(26400, 10101)
```

In [132]:
```
# Selecting top 2000 features
X_tr_new = SelectKBest(chi2, k=2000).fit_transform(X_tr, y_train)
print (X_tr_new.shape)
```

```
(35912, 2000)
```

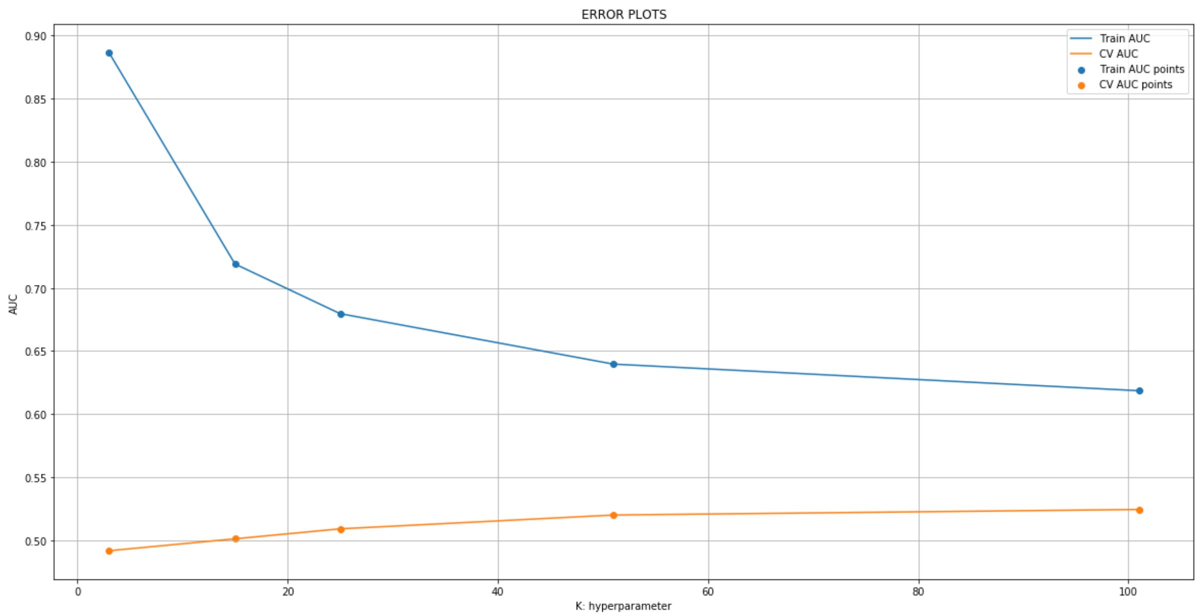
In [134]:
```
X_cr_new = SelectKBest(chi2, k=2000).fit_transform(X_cr, y_cv)
print (X_cr_new.shape)
X_test_new = SelectKBest(chi2, k=2000).fit_transform(X_te, y_test)
print (X_test_new.shape)
```

```
(17688, 2000)
(26400, 2000)
```

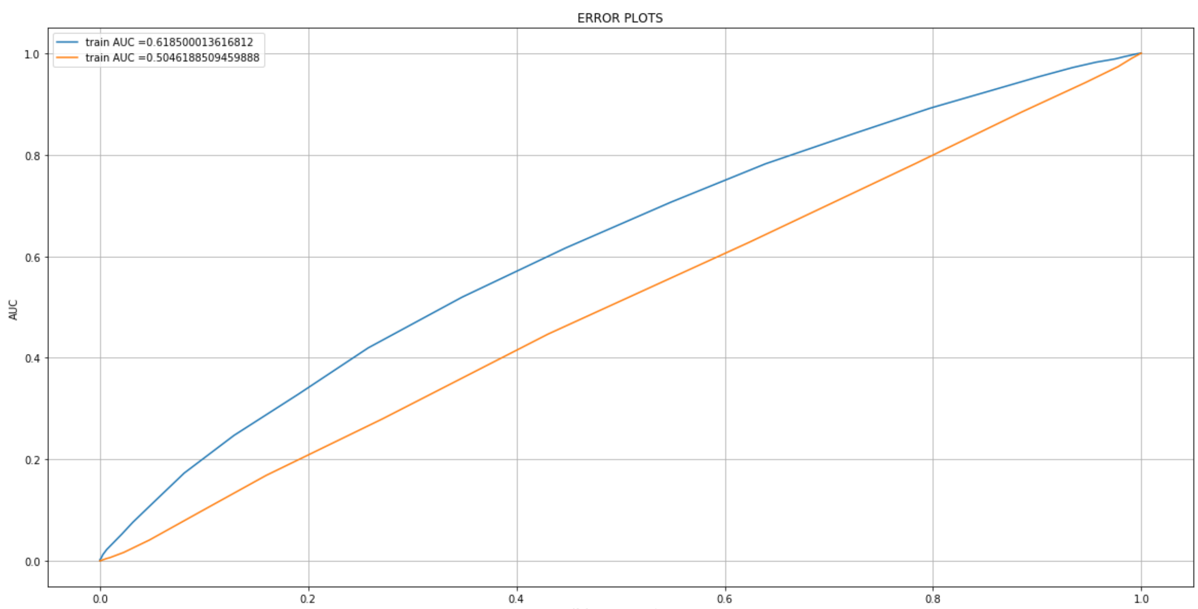
In [135]:
```
# Hyperparameter Tuning and Error plots with Best 2000 features
train_auc = []
cv_auc = []
K = [3, 15, 25, 51, 101]
for i in tqdm(K):
    neigh = KNeighborsClassifier(n_neighbors=i, n_jobs=-1)
    neigh.fit(X_tr_new, y_train)

    y_train_pred = batch_predict(neigh, X_tr_new)
    y_cv_pred = batch_predict(neigh, X_cr_new)

    # roc_auc_score(y_true, y_score) the 2nd parameter should be probability e
stimates of the positive class
    # not the predicted outputs
    train_auc.append(roc_auc_score(y_train,y_train_pred))
    cv_auc.append(roc_auc_score(y_cv, y_cv_pred))
```

```
  0%|
| 0/5 [00:00<?, ?it/s]
 20%|
| 1/5 [00:50<03:22, 50.67s/it]
 40%|
| 2/5 [01:45<02:35, 51.86s/it]
 60%|
| 3/5 [02:39<01:45, 52.59s/it]
 80%|
| 4/5 [03:33<00:53, 53.09s/it]
100%|
| 5/5 [04:29<00:00, 53.83s/it]
```

In [136]:
```
plt.figure(figsize=(20,10))
plt.plot(K, train_auc, label='Train AUC')
plt.plot(K, cv_auc, label='CV AUC')

plt.scatter(K, train_auc, label='Train AUC points')
plt.scatter(K, cv_auc, label='CV AUC points')

plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



In [138]:
```
best_k = 101
from sklearn.metrics import roc_curve, auc


neigh = KNeighborsClassifier(n_neighbors=best_k, n_jobs=-1)
neigh.fit(X_tr_new, y_train)
# roc_auc_score(y_true, y_score) the 2nd parameter should be probability estim
ates of the positive class
# not the predicted outputs

y_train_pred = batch_predict(neigh, X_tr_new)
y_test_pred = batch_predict(neigh, X_test_new)

train_fpr, train_tpr, tr_thresholds = roc_curve(y_train, y_train_pred)
test_fpr, test_tpr, te_thresholds = roc_curve(y_test, y_test_pred)
plt.figure(figsize=(20,10))
plt.plot(train_fpr, train_tpr, label="train AUC ="+str(auc(train_fpr, train_tp
r)))
plt.plot(test_fpr, test_tpr, label="test AUC ="+str(auc(test_fpr, test_tpr)))
plt.legend()
plt.xlabel("K: hyperparameter")
plt.ylabel("AUC")
plt.title("ERROR PLOTS")
plt.grid()
plt.show()
```



The Blue plot in the Graph is Train AUC and Orange Plot is Test AUC. Initially the label was accidentally copy pasted as Train AUC for both. Since redrawing plot would require to re run entire ipynb and it takes up a lot of time. so documenting the information as a markdown.

3. Conclusions

In [141]:
```
# Please compare all your models using Prettytable library
from prettytable import PrettyTable

#If you get a ModuleNotFoundError error , install prettytable using: pip3 inst
all prettytable

x = PrettyTable()
x.field_names = ["Vectorizer", "Model", "Hyperparameter", "Train AUC", "Test A
UC"]

x.add_row(["BOW","Brute",105,0.647,0.587])
x.add_row(["TFIDF","Brute",105,0.6322,0.546])
x.add_row(["W2V","Brute",101,0.620,0.548])
x.add_row(["TFIDF W2V","Brute",101,0.617,0.548])
print(x)
```

```
+------------+-------+----------------+-----------+----------+
| Vectorizer | Model | Hyperparameter | Train AUC | Test AUC |
+------------+-------+----------------+-----------+----------+
|    BOW     | Brute |      105       |   0.647   |  0.587   |
|   TFIDF    | Brute |      105       |   0.6322  |  0.546   |
|    W2V     | Brute |      101       |    0.62   |  0.548   |
| TFIDF W2V  | Brute |      101       |   0.617   |  0.548   |
+------------+-------+----------------+-----------+----------+
```

Summary: There was not a lot of difference in the AUC scores with all the features selected and top 2000 feature selection. WIth SelectionKBest the Train AUC was around 0.61 and Test AUC was around 0.50