RAMAIAH UNIVERSITY OF APPLIED SCIENCES

Assignment

Course Code	CSC402A
Course Name	Data Mining
Programme	B.Tech
Department	CSE
Faculty	FET

Name of the Student	Satyajit Ghana
Reg. No.	17ETCS002159
Semester/Year	07/2021
Course Leader(s)	Prof. Mohan Kumar

Declaration Sheet

Student Name	Satyajit Ghana		
Reg. No	17ETCS002159		
Programme	B.Tech	Semester/Year	07/2021
Course Code	CSC402A		
Course Title	Data Mining		
Course Date		to	
Course Leader	Prof. Mohan Kumar		

Declaration

The assignment submitted herewith is a result of my own investigations and that I have conformed to the guidelines against plagiarism as laid out in the Student Handbook. All sections of the text and results, which have been obtained from other sources, are fully referenced. I understand that cheating and plagiarism constitute a breach of University regulations and will be dealt with accordingly.

Signature of the Student		Date	
Submission date stamp (by Examination & Assessment Section)			

Signature of the Course Leader and date	Signature of the Reviewer and date

Contents

Declaration Sheet ii

Contents iii

1 Question 1 4

1.1 Introduction 4

1.2 Supervised Learning 7

1.3 Unsupervised Learning 9

1.4 Comparative Analysis and Conclusion 11

1.4.1 Conclusion 13

2 Appendix 14

1 Question 1

Solution to Question No. 1 Part B

1.1 Introduction

A few more pre-processing steps were performed on the dataset, to clean it, which will then be used by the learning algorithms.

Columns like [user_id, isbn, city, state] are dropped, since they don't contribute to the rating classification, and they don't really tell a lot about rating of the book (not much of information gain), also something to note (since we are considering the country) column, that most of the users (~70%) are from usa, which may/may not contribute to lot to the accuracy of the classification.

```
[ ]  1 dataset.head()
```

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	5	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

The target column is `book_rating`, now we have 10 classes for `book_rating`, but to make the task a little simpler we will be labelling the `book_rating` into 3 classes, namely `[low, mid, high]`.

The stats for them are,

high	mid	low
58.482%	39.248%	2.269%

```
[ ]  1 bins = [0, 3, 7, 10]
     2 names = ['low', 'mid', 'high']
     3
     4 dataset['book_rating'] = pd.cut(dataset['book_rating'], bins, labels=names)
```

```
[ ]  1 dataset.head()
```

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	mid	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

```
[ ]  1 dataset['book_rating'].value_counts()

     high    213208
     mid     143087
     low       8275
     Name: book_rating, dtype: int64
```

To perform any kind of learning on this dataset we would have to convert the categorical values into numerical, to do this there are two options

- OneHotEncoder
- LabelEncoder

Label Encoder is used for those categorical values which have a "ordering" associated with them, for example the temperature can be cold, warm or hot, we can assign the values 0, 1, 2 to them, that makes sense right? Yes! But now let's say the categorical values are Satyajit, Mohan, Ram how can a number be assigned to them? We cannot assign 0, 1, 2 to them, because we don't how to order them, to deal with this problem we use OneHotEncoder, which simply assigns them `[1, 0, 0]`, `[0, 1, 0]` and `[0, 0, 1]` which are basically indicating presence, of `[Satyajit, Mohan, Ram]` and there is no ordering involved here.

But looking at our dataset, we have a lot of data, and the since we are considering the book name, author, publisher all as categorical value, they will have quite a lot of categories, which

means that when we do one hot encoding, there will be a lot of columns generating, which will be very sparse ofcourse, but this means that we will be restricted by memory consumption.

```
[12]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
[13]  numeric_features = ['age', 'year_of_publication']
      numeric_transformer = Pipeline(steps=[
          ('imputer', SimpleImputer(strategy='median')),
          ('scaler', StandardScaler())])
```

```
[14]  categorical_features = ['book_title', 'book_author', 'publisher', 'country']
      categorical_transformer = OneHotEncoder(handle_unknown='ignore')
```

```
[15]  preprocessor = ColumnTransformer(
          transformers=[
              ('num', numeric_transformer, numeric_features),
              ('cat', categorical_transformer, categorical_features)])
```

```
[ ]   dummy = preprocessor.fit_transform(X_train)
      dummy.shape
```

```
(291656, 177473)
```

If we were to store all the all data, and each cell occupying 4 bytes, this is how much space we would require

```
[ ]   f'{(dummy.shape[0] * dummy.shape[1] * 4) // 1024 // 1024 // 1024} giga bytes'
```

```
'192 giga bytes'
```

From the above illustration, we can see if we try to save this data, we would approximately require 192 GB, and that is just the training set. (I did try it on a 128GB RAM machine, and I got out-of-memory error). We'll later discuss some of the solutions to this.

1.2 Supervised Learning

Supervised Learning was performed on the cleaned dataset, with OneHotEncoding and StandardScaler with the following algorithms

- Decision Tree
- SVC (Support Vector Classification)
- Passive Aggressive
- Ridge Classifier
- AdaBoost
- LinearSVC
- MLP (Multi Later Perceptron)
- RandomForest

Refer to the notebook attached at the end for more information of accuracy, confusion matrix, f1 score, etc., here we will compare the accuracies and time taken to train for all the models.

Algorithm	**F1 Score**	**Balanced Accuracy**	**Avg Precision**	**Avg Recall**	**Time Taken (s)**
DecisionTree	0.60	0.36	0.58	0.60	146
SVC	NULL	NULL	NULL	NULL	>1000
PassiveAggresive	0.55	0.37	0.55	0.55	**8**
Ridge	0.58	0.37	0.56	0.59	22
AdaBoost	0.59	0.34	0.57	0.59	50
LinearSVC	0.58	0.37	0.56	0.59	200
MLP	**0.61**	**0.38**	**0.60**	**0.61**	630
RandomForest	0.58	0.33	0.60	0.59	537

All of the above models were trained by using Imputation, and Standard Scaler, but we can see and vary the scaler to see how it affects the accuracy, to do that we consider the following scalers, and since Passive Aggressive algorithm was the faster, we will stick with that,

- StandardScaler
- MinMaxScaler
- RobustScaler
- MaxAbsScaler
- PowerTransformer
- QuantileTransfomer
- Normalizer
- UnScaled (Original Data)

Scaler	**F1 Score**	**Balanced Accuracy**	**Avg Precision**	**Avg Recall**
StandardScaler	0.561	**0.378**	0.55	0.56
MinMaxScaler	0.541	0.369	0.55	0.54
MaxAbsScaler	**0.588**	0.359	**0.55**	**0.59**
RobustScaler	0.576	0.366	0.54	0.58
PowerTransform (Yeo-Johnson)	0.525	0.374	0.55	0.53
QuantileTransform	0.519	0.371	0.54	0.52
Normalizer (L2)	0.570	0.366	0.54	0.57
UnScaled	0.583	0.364	0.55	0.58

1.3 Unsupervised Learning

In the case of unsupervised algorithms, most of them crashed on the original data, since the data was very sparse, and a lot of memory was required.

Learning from the problems faced in supervised learning, one solution to deal with text data is to perform NLP (Natural Language Processing) on the text itself, i.e. we can use pretrained word embeddings to encode the book names, authors, publishers, country into a dense matrix, which can be easily done by using spacy language models.

All of the columns in the dataset was concatenated to form a single sentence, we will later encode this sentence and try to apply clustering algorithm to predict rating.

```
[8]
c_dataset = dataset["book_title"].astype(str) + " "  + \
            dataset["book_author"].astype(str) + " " + \
            dataset["publisher"].astype(str) + " " + \
            dataset["year_of_publication"].astype(str) + " " + \
            dataset["age"].astype(str) + " " + \
            dataset["country"].astype(str)

[9]
for ex in c_dataset.sample(frac=0.2)[:5]:
    print(ex)

The Vanishing Vampire (The Accidental Monsters , No 1) David Lubar Scholastic 1997.0 12.0 usa
REMEMBER ME Mary Higgins Clark Simon &amp;amp; Schuster 1994.0 34.0 usa
The Mummy or Ramses the Damned Anne Rice Ballantine Books 1991.0 22.0 usa
Bittersweet Rain Sandra Brown Warner Books 2000.0 34.0 usa
Arthur Stephen R. Lawhead Zondervan Publishing Company 1996.0 21.0 usa
```

Also learning from the fact that I don't have enough time to fit a model on this large (300,000) data, we can simplify evaluation of algorithms by sampling a fraction of the original dataset. Even on a really good machine (i7-5930K, 128GB RAM), the model takes a lot of time to fit.

The following algorithms were used,

- KMedoids Clustering

- Agglomerative Clustering
- DBSCAN
- KMeans
- HDBSCAN
- MiniBatchKMeans

Algorithm	**F1 Score**	**Balanced Accuracy**	**Avg Precision**	**Avg Recall**	**Time Taken (s)**
KMedoids	0.30	0.31	0.50	0.31	223
Agglomerative	0.27	0.30	0.48	0.27	119
DBSCAN	**0.58**	**0.33**	0.42	**0.58**	**116**
KMeans	0.31	0.33	0.50	0.32	222
HDBSCAN	**0.58**	**0.33**	**0.54**	**0.58**	121
MiniBatch KMeans	0.33	0.32	0.49	0.34	217

To see the classification report, and the confusion matrix, see appendix.

1.4 Comparative Analysis and Conclusion

If we were to compare the supervised and unsupervised models, they have pretty much comparable accuracies (~58-61%), but one thing that cannot be compared is the execution time, since we used clustering algorithms, they require heavy memory usage, and a lot of processing time.

The highest accuracy in Supervised Learning was achieved by MLP (Multi-Layer Perceptron) of 61%, and in Unsupervised Learning, highest was achieved by HDBSCAN of 58%. And in terms of Scaler, it was observed that MaxAbsScaler gave the highest accuracy compared to other scalers.

Something even more important to address is why the accuracy is so low, the `"high"` class of `book_rating` has 58% total records, and this means that if we were to classify any input as `"high"` we have a probability of `0.58` to be correct, this is exactly like a `OneR` learner, but the reason why we are observing such bad accuracy is because there isn't much correlation between the attributes and the target value, this dataset was never meant to do something like this, it was more towards recommending books to a new user with a given past book purchase history, predicting rating based on country, book title, publisher just does not make sense, just based on the book title, how could a book be rated better? One thing that could make sense is the author, there are certain authors that have a good reputation and they have a higher chance of getting a good rating. But in this assignment, we have considered all the attributes, which could have been a mistake, one lesson learnt is that, grabbing all the available data can have negative effects to the accuracy.

But there is another issue, if we see the number of categorical values for each of the categorical columns,

```
[136] 1 dataset['book_title'].cat.categories.shape
      (132033,)

[139] 1 dataset['book_author'].cat.categories.shape
      (60652,)

[140] 1 dataset['publisher'].cat.categories.shape
      (11311,)

[141] 1 dataset['country'].cat.categories.shape
      (51,)
```

Something we can clearly observe is that, that is a lot of categories, for specially the book_title, this means that if we use OneHotEncoding, we will get those many columns added to the dataset. This does make it really difficult to train a model, scipy has a csr representation (Compressed Sparse Row Matrix) which does help to a lot of extend, but still this is the biggest issue that has to be addressed.

We can take inspiration from text classification models and NLP (Natural Language Processing), what is done for it, is that each of the words in dictionary is a sparse matrix, and then a model is used to convert this sparse matrix into a dense matrix, for example lets say the word `"the"` is represented as `[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]`, if we use a word embedding model, such as from spacy, we would get something like `[0.45673, 0.67395, 0.98729]` which is a much better representation as before, this matrix can then be used to train. This inspiration was tried in the clustering algorithms in 1.3, and it quite did not produce good results. But I would blame the dataset for that. On top of giving the dense matrix, we can further reduce the dimensionality by using SVD (Singular Value Decomposition).

I had a hunch that if I were to use GPT-2 or BERT, the model would have been really great (although it might overfit, but still), but due to time constrains, is wasn't possible, but it would be a good exploration domain.

1.4.1 Conclusion

Never rely on SciKit Learn, the algorithms are really slow (except K-Mean) and are not quite optimized enough, instead `C++` backend algorithms give much better speed.

Also, when a classification model has to be made, in which there is a lot of attributes in text in them, it would be better, to just convert those specific attributes to a dense matrix using a pretrained word embedding model, or an even better improvement can be to use `n-gram` models.

To conclude, we were not really successful in creating a good classification model for the given dataset, but we were definitely able to compare different supervised models, unsupervised models and numeric scaling techniques, overall, the outcome of this course was covered.

2 Appendix

In [ ]:
```
import gdown

url = 'https://drive.google.com/uc?id=1PL13wgXLfXcsrkKNuVIaNJdGXrIqv2mv'
output = 'book_crossing.cleaned.csv'
gdown.download(url, output, quiet=False)
```

```
Downloading...
From: https://drive.google.com/uc?id=1PL13wgXLfXcsrkKNuVIaNJdGXrIqv2mv
To: /content/book_crossing.cleaned.csv
44.9MB [00:01, 40.7MB/s]
```

Out[ ]:
```
'book_crossing.cleaned.csv'
```

In [ ]:
```
%matplotlib inline

import scipy
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

sns.set()
palette = sns.color_palette("icefire")

plt.style.use('ggplot')

sns.set_context("talk")
```

BookCrossing - Cleaning

In [ ]:
```
dataset = pd.read_csv('book_crossing.cleaned.csv')
```

In [ ]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 383849 entries, 0 to 383848
Data columns (total 11 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   user_id              383849 non-null  int64
 1   age                  383849 non-null  int64
 2   isbn                 383849 non-null  object
 3   book_rating          383849 non-null  int64
 4   book_title           383849 non-null  object
 5   book_author          383849 non-null  object
 6   year_of_publication  383849 non-null  int64
 7   publisher            383849 non-null  object
 8   city                 375223 non-null  object
 9   state                371248 non-null  object
 10  country              366406 non-null  object
dtypes: int64(4), object(7)
memory usage: 32.2+ MB
```

We won't be considering city, state, because they don't really tell a lot of the rating of a book, but also most of the users (~70%) are from usa (which may not contribute a lot to accuracy of classification, but we'll consider it), and the location is realated to the user, and not the book directly, we'll also be dropping isbn, user_id, since they dont contribute to classification of rating

In [ ]:
```
dataset = dataset.drop(['user_id', 'isbn', 'city', 'state'], axis=1)
```

In [ ]:
```
f'Dataset Shape : {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape : (383849, 7)'
```

In [ ]:
```
dataset.dropna(inplace=True)
```

In [ ]:
```
f'Dataset Shape after dropping NA: {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape after dropping NA: (366406, 7)'
```

In [ ]:
```
dataset.head()
```

Out[ ]:

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	5	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	8	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	9	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

In [ ]:
```
dataset.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	max
age	366406.0	35.860998	10.448608	5.0	30.0	34.0	40.0	100.0
book_rating	366406.0	7.635975	1.836354	1.0	7.0	8.0	9.0	10.0
year_of_publication	366406.0	1995.670314	7.397156	1376.0	1993.0	1997.0	2001.0	2006.0

We'll remove the rows which have a country which has value count <= 50

In [ ]:
```
dataset = dataset.groupby('country').filter(lambda x: len(x) > 50)
```

In [ ]:
```
dataset.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	75%	max
age	364570.0	35.867227	10.447887	5.0	30.0	34.0	40.0	100.0
book_rating	364570.0	7.636709	1.835857	1.0	7.0	8.0	9.0	10.0
year_of_publication	364570.0	1995.667164	7.400552	1376.0	1993.0	1997.0	2001.0	2006.0

In [ ]:
```
f'Dataset Shape : {dataset.shape}'
```

Out[ ]:
```
'Dataset Shape : (364570, 7)'
```

In [ ]:
```
f'Column Names: {dataset.columns.to_list()}'
```

Out[ ]:
```
"Column Names: ['age', 'book_rating', 'book_title', 'book_author', 'year_of_publication', 'publisher', 'country']"
```

In [ ]:
```
dataset['book_rating'].value_counts()
```

Out[ ]:
```
8     87090
10    68038
7     63036
9     58080
5     42988
6     29943
4      7120
3      4746
2      2198
1      1331
Name: book_rating, dtype: int64
```

We'll now convert the rating into classification categories

In [ ]:
```
bins = [0, 3, 7, 10]
names = ['low', 'mid', 'high']

dataset['book_rating'] = pd.cut(dataset['book_rating'], bins, labels=names)
```

In [ ]:
```
dataset.head()
```

Out[ ]:

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	34	mid	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
2	30	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
4	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
5	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada
6	34	high	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	canada

In [ ]:
```
dataset['book_rating'].value_counts()
```

Out[ ]:
```
high    213208
mid     143087
low       8275
Name: book_rating, dtype: int64
```

In [ ]:
```
dataset.to_csv('book_crossing.classification.cleaned.csv', index=False)
```

[0] !ls -l /kaggle/input/cifar-10/*.7z
```
-rw-r--r-- 1 root root 151851130 Jun 17 2020 /kaggle/input/cifar-10/cifar-10.7z
```

Book Crossing - Classification

[1] !apt-get install p7zip
```
Reading package lists...
Building dependency tree...
Reading state information...
The following NEW packages will be installed:
  p7zip-full
0 upgraded, 1 newly installed, 0 to remove and 110 not upgraded.
Need to get 1,630 kB of archives.
After this operation, 5,035 kB of additional disk space will be used.
Get:1 http://deb.debian.org/debian/ bookworm/main amd64 p7zip-full amd64 16.02+dfsg-8 [1,630 kB]
Fetched 1,630 kB in 1s (1,164 kB/s)
Selecting previously unselected package p7zip-full.
(Reading database ... 124,947 files and directories currently installed.)
Preparing to unpack .../p7zip-full_16.02+dfsg-8_amd64.deb ...
Unpacking p7zip-full (16.02+dfsg-8) ...
Setting up p7zip-full (16.02+dfsg-8) ...
Processing triggers for man-db (2.9.4-2) ...
```

[2] !7z x -o/kaggle/working -y /kaggle/input/cifar-10/cifar-10.7z | tail -20
```
Everything is Ok

Folders: 6
Files: 300000
Size:       1342177280
Compressed: 151851130
```

[3] !ls -lR /kaggle/working | head -40
```
/kaggle/working:
total 20
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images_test
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images_train
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images_validation
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images_validation_with_labels
drwxr-xr-x 2 root root 4096 Jun 17 2020 cifar10_images_with_labels

/kaggle/working/cifar10_images:
total 0
drwxr-xr-x 2 root root 4096 Jun 17 2020 airplane
drwxr-xr-x 2 root root 4096 Jun 17 2020 automobile
drwxr-xr-x 2 root root 4096 Jun 17 2020 bird
drwxr-xr-x 2 root root 4096 Jun 17 2020 cat
drwxr-xr-x 2 root root 4096 Jun 17 2020 deer
drwxr-xr-x 2 root root 4096 Jun 17 2020 dog
drwxr-xr-x 2 root root 4096 Jun 17 2020 frog
drwxr-xr-x 2 root root 4096 Jun 17 2020 horse
drwxr-xr-x 2 root root 4096 Jun 17 2020 ship
drwxr-xr-x 2 root root 4096 Jun 17 2020 truck

/kaggle/working/cifar10_images/airplane:
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0005.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0006.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0007.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0008.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0009.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0010.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0011.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0012.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0013.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0014.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0015.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0016.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0017.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0018.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0019.jpg -> ../../cifar10_images_train
```

[4] !ls -l /kaggle/working/cifar10_images/airplane | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[5] !ls -l /kaggle/working/cifar10_images/airplane | wc -l
```
5001
```

[6] !ls -l /kaggle/working/cifar10_images/airplane | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[7] !ls -l /kaggle/working/cifar10_images/bird | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[8] !ls -l /kaggle/working/cifar10_images/bird | wc -l
```
5001
```

[9] !ls -l /kaggle/working/cifar10_images/bird | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[10] !ls -l /kaggle/working/cifar10_images/cat | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[11] !ls -l /kaggle/working/cifar10_images/cat | wc -l
```
5001
```

[12] !ls -l /kaggle/working/cifar10_images/cat | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[13] !ls -l /kaggle/working/cifar10_images_test | head -5
```
total 0
drwxr-xr-x 2 root root 4096 Jun 17 2020 airplane
drwxr-xr-x 2 root root 4096 Jun 17 2020 automobile
drwxr-xr-x 2 root root 4096 Jun 17 2020 bird
drwxr-xr-x 2 root root 4096 Jun 17 2020 cat
drwxr-xr-x 2 root root 4096 Jun 17 2020 deer
```

[14] !ls -l /kaggle/working/cifar10_images_test/airplane | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[15] !ls -l /kaggle/working/cifar10_images_test/airplane | wc -l
```
5001
```

[16] !ls -l /kaggle/working/cifar10_images_test/airplane | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[17] !ls -l /kaggle/working/cifar10_images_validation | head -5
```
total 0
drwxr-xr-x 2 root root 4096 Jun 17 2020 airplane
drwxr-xr-x 2 root root 4096 Jun 17 2020 automobile
drwxr-xr-x 2 root root 4096 Jun 17 2020 bird
drwxr-xr-x 2 root root 4096 Jun 17 2020 cat
drwxr-xr-x 2 root root 4096 Jun 17 2020 deer
```

[18] !ls -l /kaggle/working/cifar10_images_validation/airplane | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[19] !ls -l /kaggle/working/cifar10_images_validation/airplane | wc -l
```
5001
```

[20] !ls -l /kaggle/working/cifar10_images_validation/airplane | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[21] !ls -l /kaggle/working/cifar10_images_validation_with_labels | head -5
```
total 0
drwxr-xr-x 2 root root 4096 Jun 17 2020 airplane
drwxr-xr-x 2 root root 4096 Jun 17 2020 automobile
drwxr-xr-x 2 root root 4096 Jun 17 2020 bird
drwxr-xr-x 2 root root 4096 Jun 17 2020 cat
drwxr-xr-x 2 root root 4096 Jun 17 2020 deer
```

[22] !ls -l /kaggle/working/cifar10_images_validation_with_labels/airplane | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[23] !ls -l /kaggle/working/cifar10_images_validation_with_labels/airplane | wc -l
```
5001
```

[24] !ls -l /kaggle/working/cifar10_images_validation_with_labels/airplane | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

[25] !ls -l /kaggle/working/cifar10_images_with_labels | head -5
```
total 0
drwxr-xr-x 2 root root 4096 Jun 17 2020 airplane
drwxr-xr-x 2 root root 4096 Jun 17 2020 automobile
drwxr-xr-x 2 root root 4096 Jun 17 2020 bird
drwxr-xr-x 2 root root 4096 Jun 17 2020 cat
drwxr-xr-x 2 root root 4096 Jun 17 2020 deer
```

[26] !ls -l /kaggle/working/cifar10_images_with_labels/airplane | head -5
```
total 0
lrwxrwxrwx 1 root root 30 Jun 17 2020 0000.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0001.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0002.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0003.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 0004.jpg -> ../../cifar10_images_train
```

[27] !ls -l /kaggle/working/cifar10_images_with_labels/airplane | wc -l
```
5001
```

[28] !ls -l /kaggle/working/cifar10_images_with_labels/airplane | tail -5
```
lrwxrwxrwx 1 root root 30 Jun 17 2020 4996.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4997.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4998.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 4999.jpg -> ../../cifar10_images_train
lrwxrwxrwx 1 root root 30 Jun 17 2020 5000.jpg -> ../../cifar10_images_train
```

Supervised Models

- Decision Tree
- Naive Bayes
- Logistic Regression
- Support Vector Machine
- Random Forest
- Gradient Boosting
- Linear SVC
- Ridge Classifier
- AdaBoost Classifier
- Linear SVC
- SGD Classifier
- MLP Classifier
- RuleFit/Fast Classifier
- Computing Scores
- Decision Tree
- Naive Bayes
- Logistic Regression
- Support Vector Machine
- Random Forest
- Gradient Boosting
- Linear SVC
- Ridge Classifier
- AdaBoost Classifier
- Linear SVC
- SGD Classifier
- MLP Classifier
- RuleFit/Fast Classifier

[29] !pip install -q -U keras-tuner tensorflow-datasets tqdm matplotlib pandas numpy scikit-learn

[30] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[31] !find /kaggle/working -maxdepth 2 -type d | head -20
```
/kaggle/working
/kaggle/working/cifar10_images
/kaggle/working/cifar10_images_test
/kaggle/working/cifar10_images_train
/kaggle/working/cifar10_images_validation
/kaggle/working/cifar10_images_validation_with_labels
/kaggle/working/cifar10_images_with_labels
```

[32] !find /kaggle/working/cifar10_images -maxdepth 2 -type d -name '*' | head -20
```
/kaggle/working/cifar10_images
/kaggle/working/cifar10_images/airplane
/kaggle/working/cifar10_images/automobile
/kaggle/working/cifar10_images/bird
/kaggle/working/cifar10_images/cat
/kaggle/working/cifar10_images/deer
/kaggle/working/cifar10_images/dog
/kaggle/working/cifar10_images/frog
/kaggle/working/cifar10_images/horse
/kaggle/working/cifar10_images/ship
/kaggle/working/cifar10_images/truck
```

[33] !find /kaggle/working/cifar10_images/airplane -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images/airplane/0000.jpg
/kaggle/working/cifar10_images/airplane/0001.jpg
/kaggle/working/cifar10_images/airplane/0002.jpg
/kaggle/working/cifar10_images/airplane/0003.jpg
/kaggle/working/cifar10_images/airplane/0004.jpg
```

[34] !find /kaggle/working/cifar10_images/airplane -type f -name '*.jpg' | wc -l
```
5001
```

[35] !find /kaggle/working/cifar10_images/airplane -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images/airplane/4996.jpg
/kaggle/working/cifar10_images/airplane/4997.jpg
/kaggle/working/cifar10_images/airplane/4998.jpg
/kaggle/working/cifar10_images/airplane/4999.jpg
/kaggle/working/cifar10_images/airplane/5000.jpg
```

[36] !find /kaggle/working/cifar10_images/bird -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images/bird/0000.jpg
/kaggle/working/cifar10_images/bird/0001.jpg
/kaggle/working/cifar10_images/bird/0002.jpg
/kaggle/working/cifar10_images/bird/0003.jpg
/kaggle/working/cifar10_images/bird/0004.jpg
```

[37] !find /kaggle/working/cifar10_images/bird -type f -name '*.jpg' | wc -l
```
5001
```

[38] !find /kaggle/working/cifar10_images/bird -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images/bird/4996.jpg
/kaggle/working/cifar10_images/bird/4997.jpg
/kaggle/working/cifar10_images/bird/4998.jpg
/kaggle/working/cifar10_images/bird/4999.jpg
/kaggle/working/cifar10_images/bird/5000.jpg
```

[39] !find /kaggle/working/cifar10_images/cat -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images/cat/0000.jpg
/kaggle/working/cifar10_images/cat/0001.jpg
/kaggle/working/cifar10_images/cat/0002.jpg
/kaggle/working/cifar10_images/cat/0003.jpg
/kaggle/working/cifar10_images/cat/0004.jpg
```

[40] !find /kaggle/working/cifar10_images/cat -type f -name '*.jpg' | wc -l
```
5001
```

[41] !find /kaggle/working/cifar10_images/cat -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images/cat/4996.jpg
/kaggle/working/cifar10_images/cat/4997.jpg
/kaggle/working/cifar10_images/cat/4998.jpg
/kaggle/working/cifar10_images/cat/4999.jpg
/kaggle/working/cifar10_images/cat/5000.jpg
```

[42] !find /kaggle/working/cifar10_images_test -maxdepth 2 -type d | head -20
```
/kaggle/working/cifar10_images_test
/kaggle/working/cifar10_images_test/airplane
/kaggle/working/cifar10_images_test/automobile
/kaggle/working/cifar10_images_test/bird
/kaggle/working/cifar10_images_test/cat
/kaggle/working/cifar10_images_test/deer
```

[43] !find /kaggle/working/cifar10_images_test/airplane -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images_test/airplane/0000.jpg
/kaggle/working/cifar10_images_test/airplane/0001.jpg
/kaggle/working/cifar10_images_test/airplane/0002.jpg
/kaggle/working/cifar10_images_test/airplane/0003.jpg
/kaggle/working/cifar10_images_test/airplane/0004.jpg
```

[44] !find /kaggle/working/cifar10_images_test/airplane -type f -name '*.jpg' | wc -l
```
5001
```

[45] !find /kaggle/working/cifar10_images_test/airplane -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images_test/airplane/4996.jpg
/kaggle/working/cifar10_images_test/airplane/4997.jpg
/kaggle/working/cifar10_images_test/airplane/4998.jpg
/kaggle/working/cifar10_images_test/airplane/4999.jpg
/kaggle/working/cifar10_images_test/airplane/5000.jpg
```

[46] !find /kaggle/working/cifar10_images_validation -maxdepth 2 -type d | head -20
```
/kaggle/working/cifar10_images_validation
/kaggle/working/cifar10_images_validation/airplane
/kaggle/working/cifar10_images_validation/automobile
/kaggle/working/cifar10_images_validation/bird
/kaggle/working/cifar10_images_validation/cat
/kaggle/working/cifar10_images_validation/deer
```

[47] !find /kaggle/working/cifar10_images_validation/airplane -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images_validation/airplane/0000.jpg
/kaggle/working/cifar10_images_validation/airplane/0001.jpg
/kaggle/working/cifar10_images_validation/airplane/0002.jpg
/kaggle/working/cifar10_images_validation/airplane/0003.jpg
/kaggle/working/cifar10_images_validation/airplane/0004.jpg
```

[48] !find /kaggle/working/cifar10_images_validation/airplane -type f -name '*.jpg' | wc -l
```
5001
```

[49] !find /kaggle/working/cifar10_images_validation/airplane -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images_validation/airplane/4996.jpg
/kaggle/working/cifar10_images_validation/airplane/4997.jpg
/kaggle/working/cifar10_images_validation/airplane/4998.jpg
/kaggle/working/cifar10_images_validation/airplane/4999.jpg
/kaggle/working/cifar10_images_validation/airplane/5000.jpg
```

[50] !find /kaggle/working/cifar10_images_validation_with_labels -maxdepth 2 -type d | head -20
```
/kaggle/working/cifar10_images_validation_with_labels
/kaggle/working/cifar10_images_validation_with_labels/airplane
/kaggle/working/cifar10_images_validation_with_labels/automobile
/kaggle/working/cifar10_images_validation_with_labels/bird
/kaggle/working/cifar10_images_validation_with_labels/cat
/kaggle/working/cifar10_images_validation_with_labels/deer
```

[51] !find /kaggle/working/cifar10_images_validation_with_labels/airplane -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images_validation_with_labels/airplane/0000.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/0001.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/0002.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/0003.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/0004.jpg
```

[52] !find /kaggle/working/cifar10_images_validation_with_labels/airplane -type f -name '*.jpg' | wc -l
```
5001
```

[53] !find /kaggle/working/cifar10_images_validation_with_labels/airplane -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images_validation_with_labels/airplane/4996.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/4997.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/4998.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/4999.jpg
/kaggle/working/cifar10_images_validation_with_labels/airplane/5000.jpg
```

[54] !find /kaggle/working/cifar10_images_with_labels -maxdepth 2 -type d | head -20
```
/kaggle/working/cifar10_images_with_labels
/kaggle/working/cifar10_images_with_labels/airplane
/kaggle/working/cifar10_images_with_labels/automobile
/kaggle/working/cifar10_images_with_labels/bird
/kaggle/working/cifar10_images_with_labels/cat
/kaggle/working/cifar10_images_with_labels/deer
```

[55] !find /kaggle/working/cifar10_images_with_labels/airplane -type f -name '*.jpg' | head -5
```
/kaggle/working/cifar10_images_with_labels/airplane/0000.jpg
/kaggle/working/cifar10_images_with_labels/airplane/0001.jpg
/kaggle/working/cifar10_images_with_labels/airplane/0002.jpg
/kaggle/working/cifar10_images_with_labels/airplane/0003.jpg
/kaggle/working/cifar10_images_with_labels/airplane/0004.jpg
```

[56] !find /kaggle/working/cifar10_images_with_labels/airplane -type f -name '*.jpg' | wc -l
```
5001
```

[57] !find /kaggle/working/cifar10_images_with_labels/airplane -type f -name '*.jpg' | tail -5
```
/kaggle/working/cifar10_images_with_labels/airplane/4996.jpg
/kaggle/working/cifar10_images_with_labels/airplane/4997.jpg
/kaggle/working/cifar10_images_with_labels/airplane/4998.jpg
/kaggle/working/cifar10_images_with_labels/airplane/4999.jpg
/kaggle/working/cifar10_images_with_labels/airplane/5000.jpg
```

[58] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[59] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[60] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[61] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[62] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[63] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[64] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[65] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[66] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[67] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[68] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[69] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[70] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[71] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[72] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[73] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[74] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[75] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[76] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[77] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[78] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[79] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[80] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[81] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[82] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PIL, sklearn, tensorflow as tf, keras_tuner as kt, tensorflow_datasets as tfds
```
TensorFlow 2.19.0 | Keras 3.10.0 | KerasTuner 1.4.8 | TFDS 4.9.9
```

[83] import os, sys, glob, pathlib, random, time, gc, warnings, numpy as np, pandas as pd, matplotlib.pyplot as plt, tqdm, tqdm.notebook as tqdm_nb, PI

In [1]: ![ ] pip install -upgrade scikit-learn

In [2]: ![ ] pip install scikit-learn-extra

In [3]: ![ ] pip install hdbscan

In [4]: ![ ] python -m keep download ucimlrepo

Book Crossing - Classification (Unsupervised)

In [5]: %matplotlib inline

```
import os
import numpy as np
import matplotlib as mpl
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

np.random.
np.random.seed(42)
%load_ext autoreload
%autoreload 2

plt.style.use('ggplot')

np.set_printoptions(threshold=20,
precision=5,
suppress=True)
```

In [6]: dataset = pd.read_csv('book_crossing.classification.clean.csv')

In [7]: dataset['age'] = dataset['age'].astype('Int64')
dataset['book_rating'] = dataset['book_rating'].astype('category')
dataset['book_rating_text'] = dataset['book_rating_text'].astype('category')
dataset['book_author'] = dataset['book_author'].astype('category')
dataset['publication_country'] = dataset['publication_country'].astype('category')
dataset['publisher'] = dataset['publisher'].astype('category')
dataset['country'] = dataset['country'].astype('category')

In [8]: dataset['book_title'].cat.categories.shape

Out[8]: (131063,)

In [9]: dataset['publisher'].cat.categories.shape

Out[9]: (13151,)

In [10]: dataset['country'].cat.categories.shape

Out[10]: (21,)

In [11]: dataset.head()

Out[11]:

	age	book_rating	book_title	book_author	year_of_publication	publisher	country
0	241	high	Classmates: Bonaventure High	2003	2000.0	HarperCollins Canada	Canada
1	241	high	Classmates: Bonaventure High	2003	2000.0	HarperCollins Canada	Canada
2	241	high	Classmates: Bonaventure High	2003	2000.0	HarperCollins Canada	Canada
3	241	high	Classmates: Bonaventure High	2003	2000.0	HarperCollins Canada	Canada
4	241	high	Classmates: Bonaventure High	2003	2000.0	HarperCollins Canada	Canada

In [12]: dataset.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 384878 entries, 0 to 384877
Data columns (total 7 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   age                  384878 non-null  Int64
 1   book_rating          384878 non-null  category
 2   book_title           384878 non-null  category
 3   book_author          384878 non-null  category
 4   year_of_publication  384878 non-null  category
 5   publisher            384878 non-null  category
 6   country              384878 non-null  category
dtypes: Int64(1), category(6)
memory usage: 15.1 MB
```

In [13]: dataset = dataset[~dataset.isin([-1,
dataset[~dataset[['publication_country']].isin([
dataset[~dataset[['publication_country']].isin([
dataset[~dataset[['publication_country']].isin([