SC1015: GENRE-RATOR

SC12 - A Multilabel Classifier of Genres

- Nathaniel Chin Yi Xuan [redacted]
- Marcus Soh Yi Qing [redacted]
- Tan Yan Chi [redacted]

https://github.com/natisaver/GoodReads-Multilabel-Genre-Prediction

TABLE OF CONTENTS

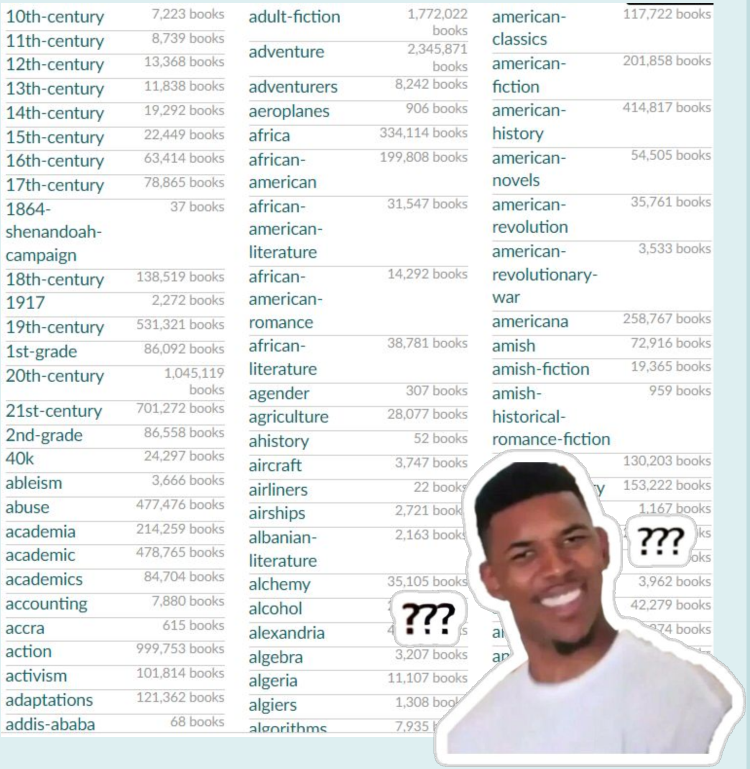
PROBLEM FORMULATION

Real-Life Problem:

- Over **900** genres on Goodreads website => **Unstandardized** definition of different genres
- Confusing for user navigation

Project Resolution:

- Aims to produce an algorithm that is able to predict the combination of genres of a book given its **plot description** and **features** to create **standardisation**

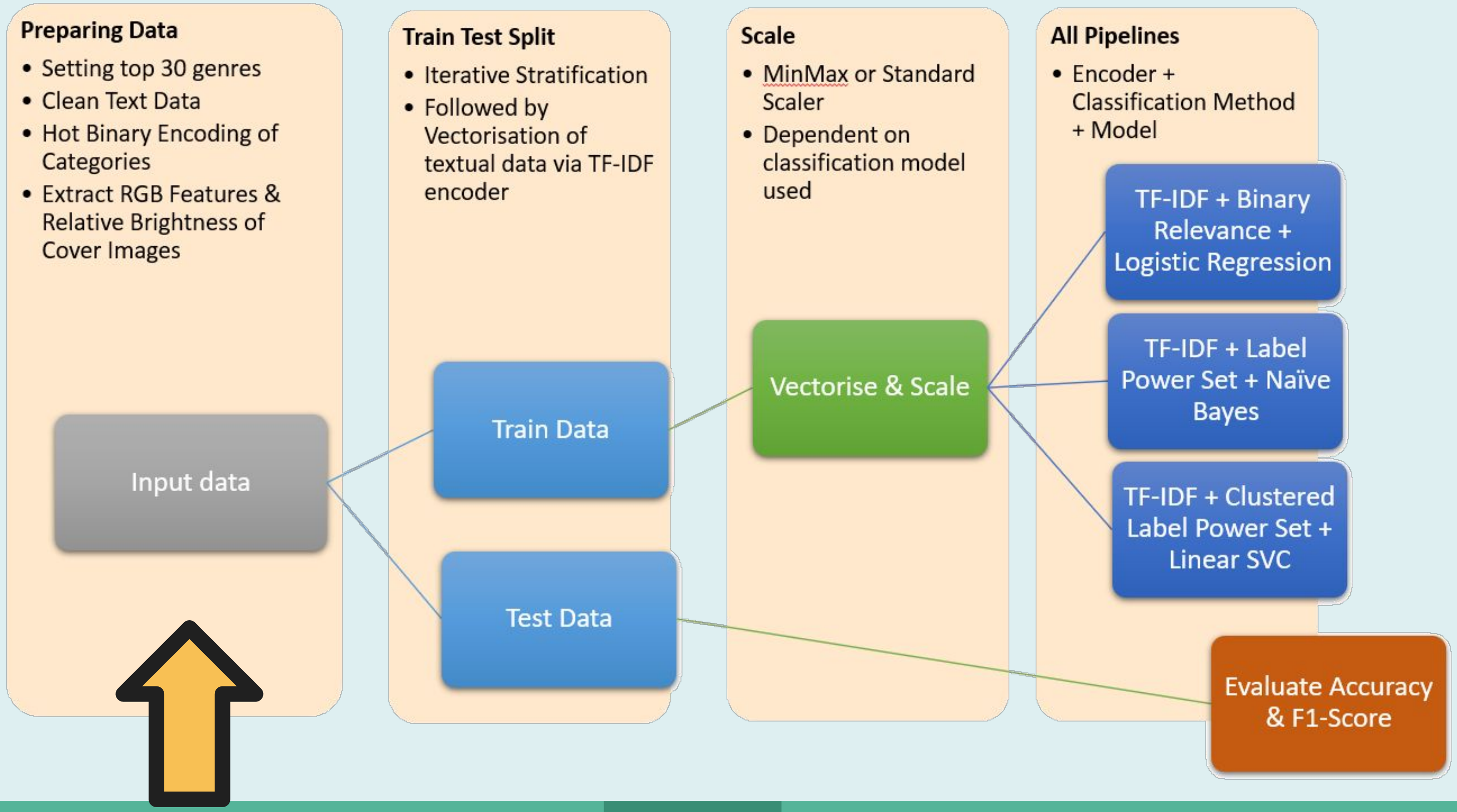


DATASET

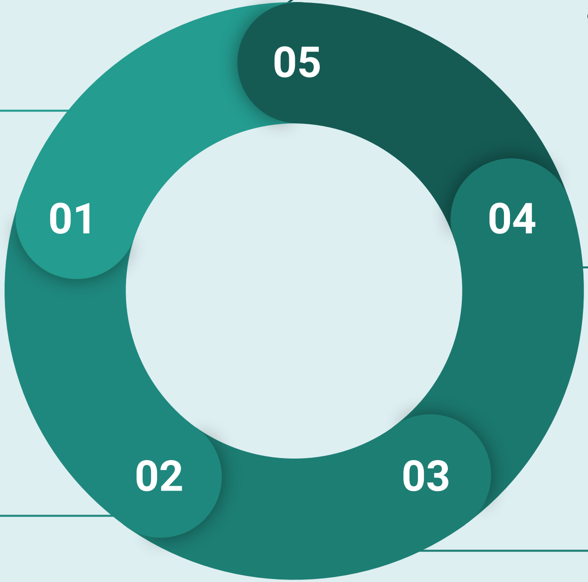
The Zenodo Dataset we have chosen contains the following columns.

```
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   bookId            52478 non-null  object
 1   title             52478 non-null  object
 2   series            23470 non-null  object
 3   author            52478 non-null  object
 4   rating            52478 non-null  float64
 5   description       51140 non-null  object
 6   language          48672 non-null  object
 7   isbn              52478 non-null  object
 8   genres            52478 non-null  object
 9   characters        52478 non-null  object
 10  bookFormat        51005 non-null  object
 11  edition           4955 non-null   object
 12  pages             50131 non-null  object
 13  publisher         48782 non-null  object
 14  publishDate       51598 non-null  object
 15  firstPublishDate  31152 non-null  object
 16  awards            52478 non-null  object
 17  numRatings        52478 non-null  int64
 18  ratingsByStars    52478 non-null  object
 19  likedPercent      51856 non-null  float64
 20  setting           52478 non-null  object
 21  coverImg          51873 non-null  object
 22  bbeScore          52478 non-null  int64
 23  bbeVotes          52478 non-null  int64
 24  price             38113 non-null  object
```

PROJECT PROCESS OVERVIEW



DATA CLEANING



Removing Non-English Books

- To ensure that the description of the books are **purely English content**
- Ensuring **consistency** when analysing textual data.

Dropping columns that are irrelevant

- Columns like ‘price’, ‘publisher’ contain too many **NULL** values to be useful for analysis
- Columns like ‘language’, ‘bookFormat’ are **irrelevant** to predicting ‘genres’

Cleaning Description Column

- Cleaning NULL
- Lemmatizing
- Stop Words
- Accented Characters
- Punctuations
- Lower Case

Cleaning Genre Column

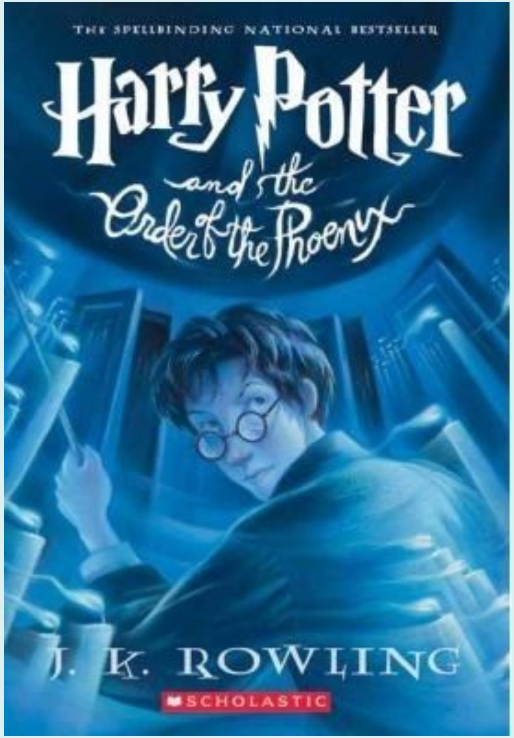
- Reducing the initial **967** “genres” to fixed **top 30 genres**

Multi-Hot Binary Encoding of Genre

- Each of the 30 genres were encoded to a **one-hot binary representation**
- If a book belongs to a genre, the value is 1 ("hot") else 0

PROCESSING COVER IMAGE

- To get additional features, we did image processing to extract the **Red, Green, Blue** Values of the cover images for all the books
- Luminance was then calculated using the formula:
  - ```
    math.sqrt(0.241*(row.r**2) + 0.691*(row.g**2) + 0.068*(row.b**2))
    ```


    ```

MODIFIED GENRE LIST

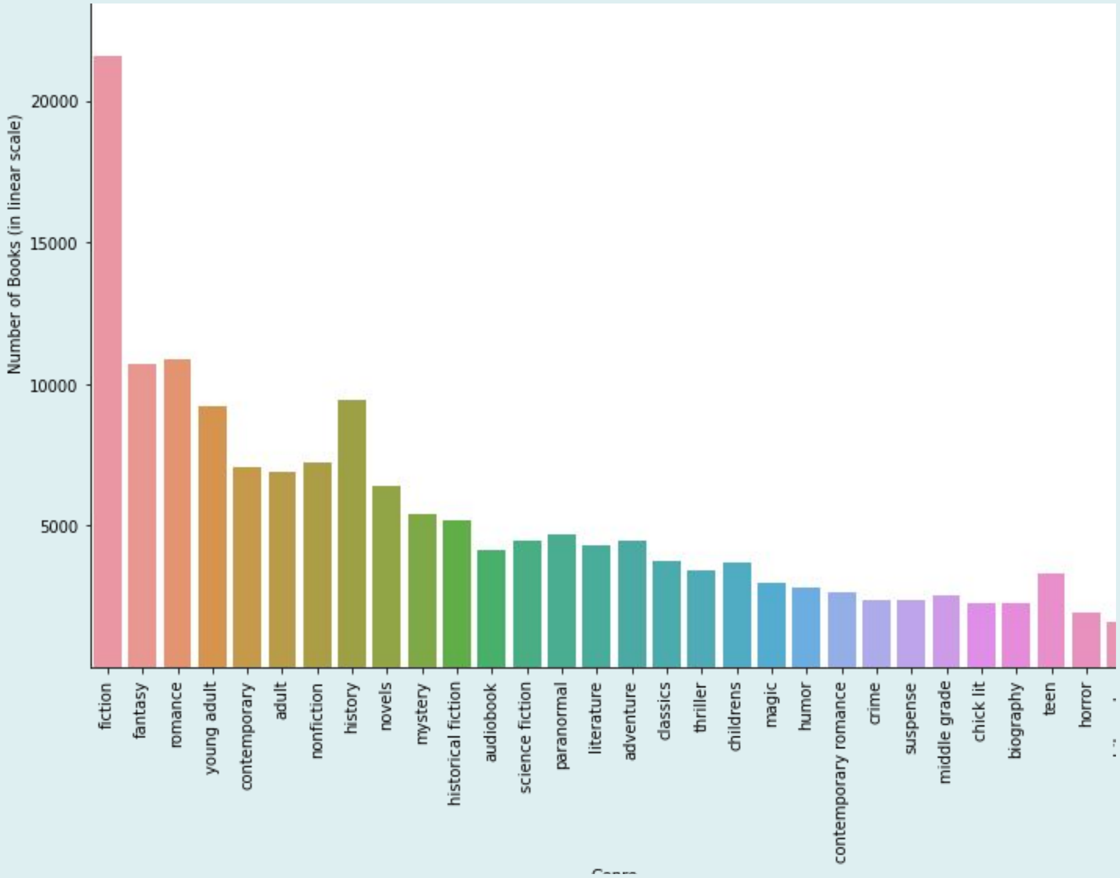
Our team manually analysed the top 60 genres and reduced them to a "top 30 genre list"

- **Purpose**
  - Standardisation
  - Combine overlapped genres
- **Example**
  - History, 11th Century, Historical → History
  - Adult Fiction → Adult and Fiction

```
top30genrelist = ['fiction',
'fantasy',
'romance',
'young adult',
'contemporary',
'adult',
'nonfiction',
'history',
'novels',
'mystery',
'historical fiction',
'audiobook',
'science fiction',
'paranormal',
'literature',
'adventure',
'classics',
'thriller',
'childrens',
'magic',
'humor',
'contemporary romance',
'crime',
'suspense',
'middle grade',
'chick lit',
'biography',
'teen',
'horror',
'philosophy']
```

EXPLORATORY DATA ANALYSIS

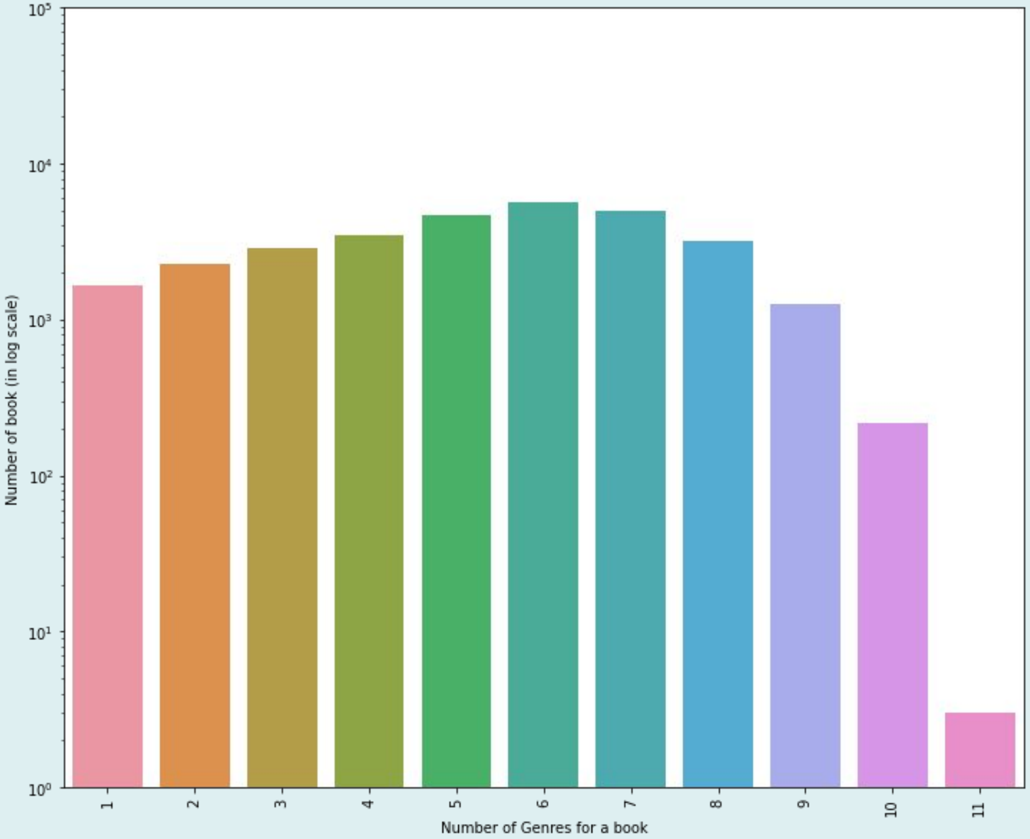
EDA - NUMBER OF BOOKS PER GENRE



Key Observations

- **Fiction** has the highest number of books with **21590**
- Followed by **Romance** which is less than half of fiction at **10862**
- The genre with the least number of books is **philosophy** with **1582** books

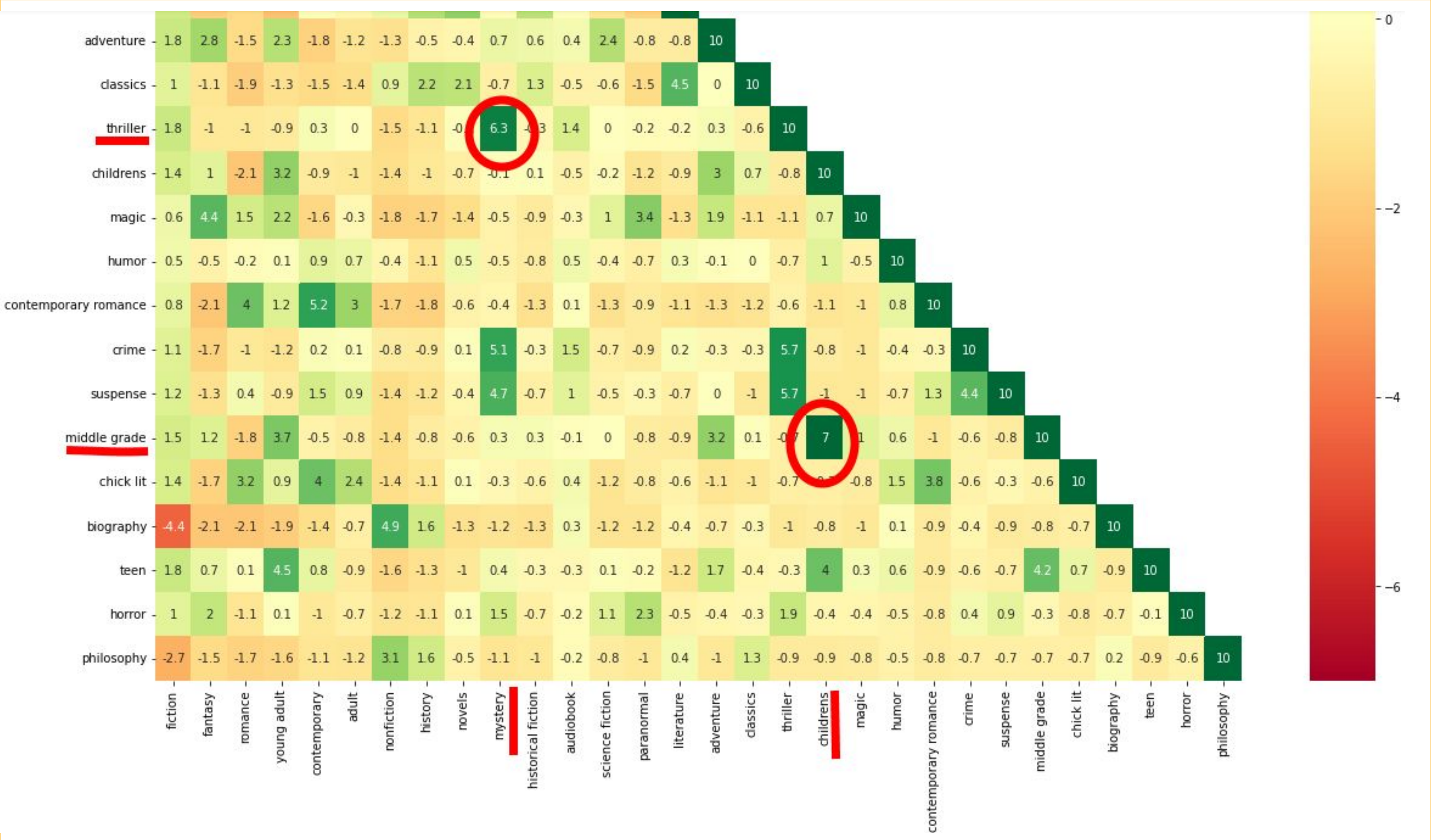
EDA - NUMBER OF GENRE PER BOOK



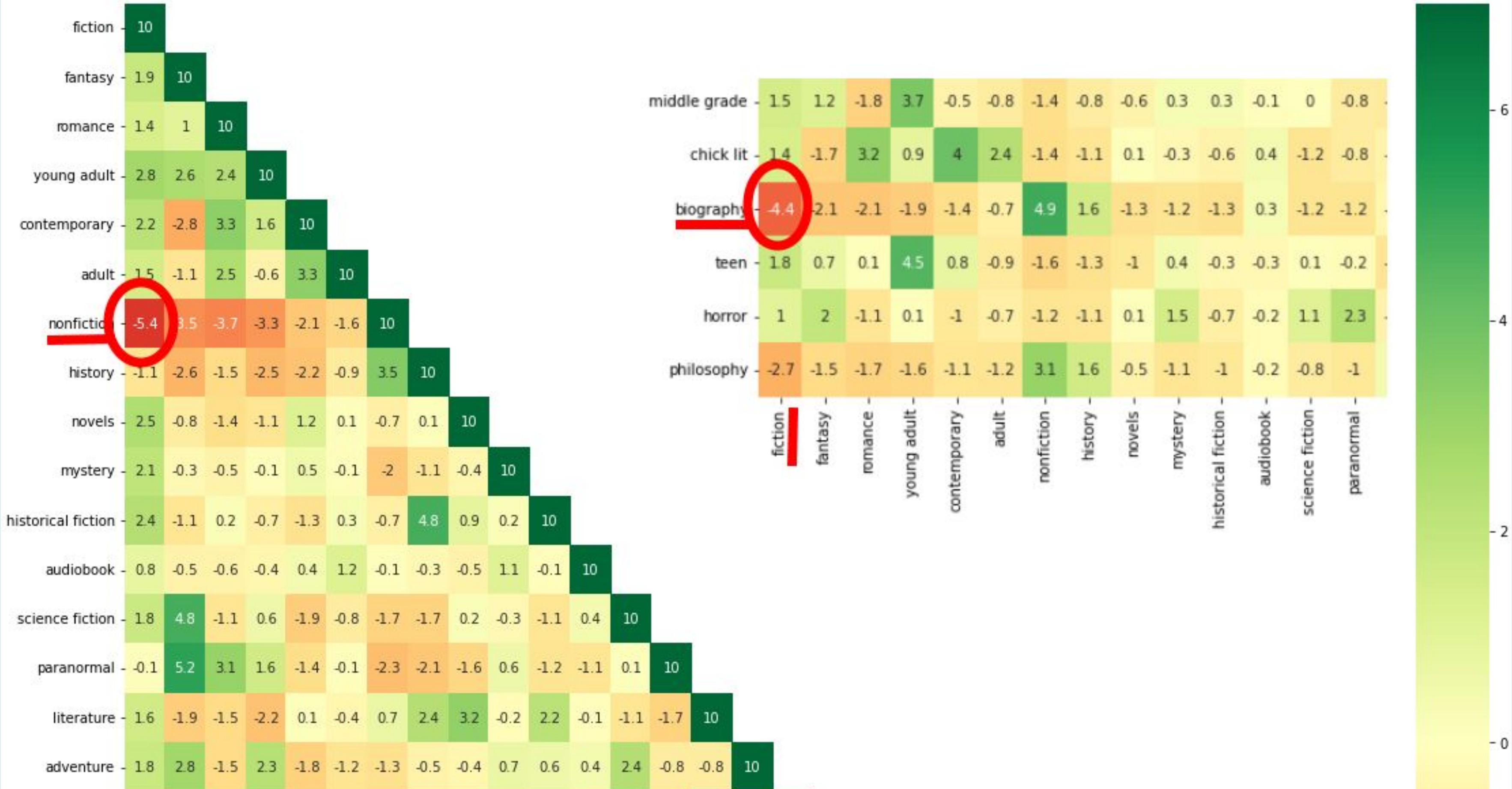
Key Observations

- On average, books have **5.28** genres.
- There are **3 books** that is associated with **11** total different genres!

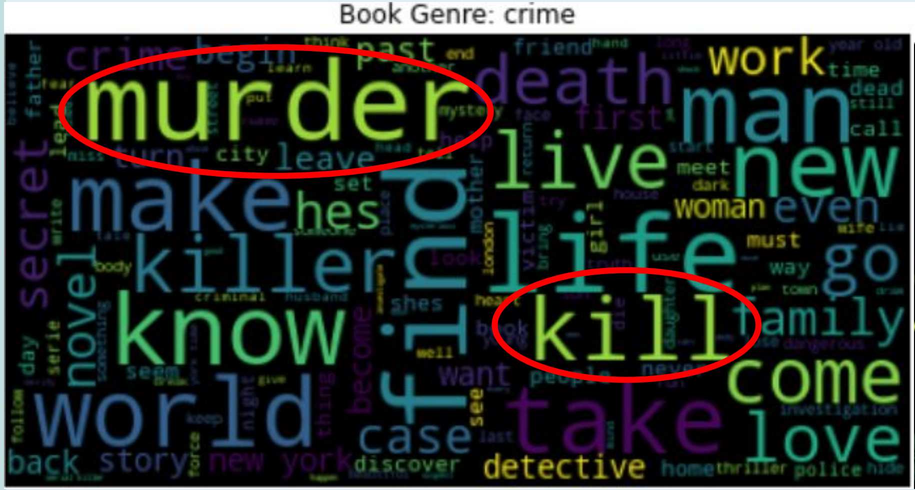
EDA - NUMBER OF GENRE PER BOOK

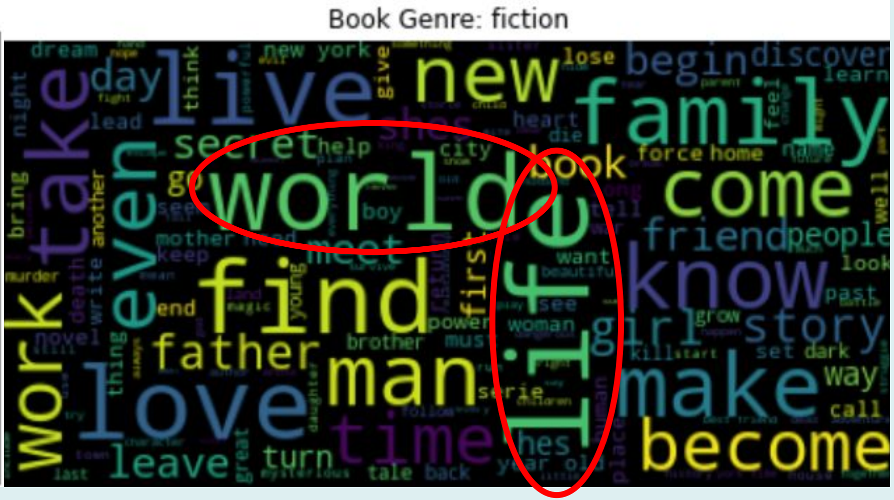


EDA - NUMBER OF GENRE PER BOOK



EXPLORATORY DATA ANALYSIS

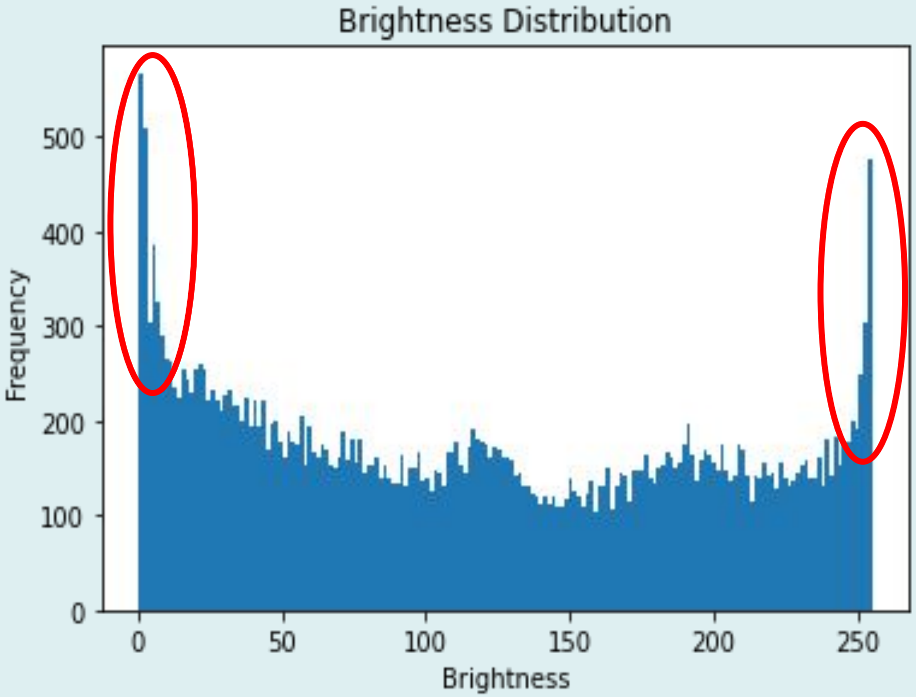




Key Observations

Certain words such as “**kill**”, “**murder**” appears in Crime, Mystery, Suspense and Thriller
“**world**” and “**life**” seems to be the most common words that appear in book descriptions

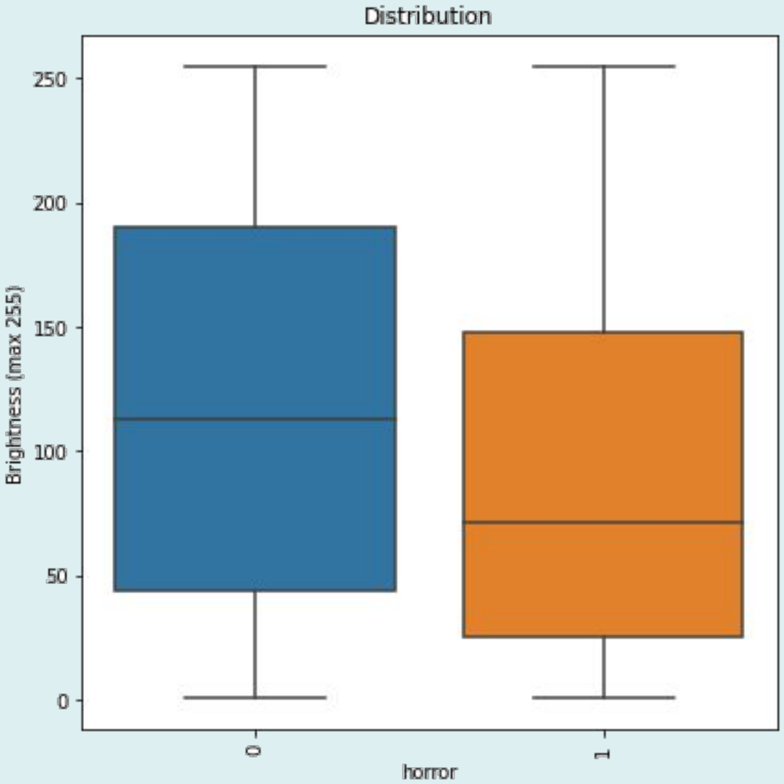
EDA - BRIGHTNESS

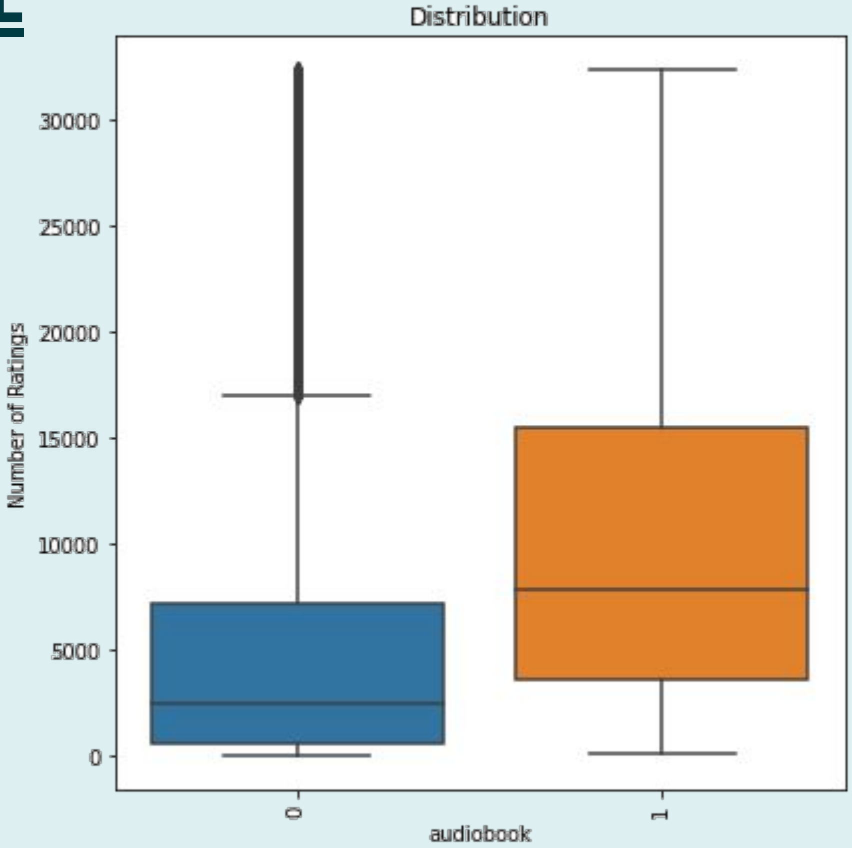


Key Observations

- Book covers tend to have **very high brightness** or **very low brightness**

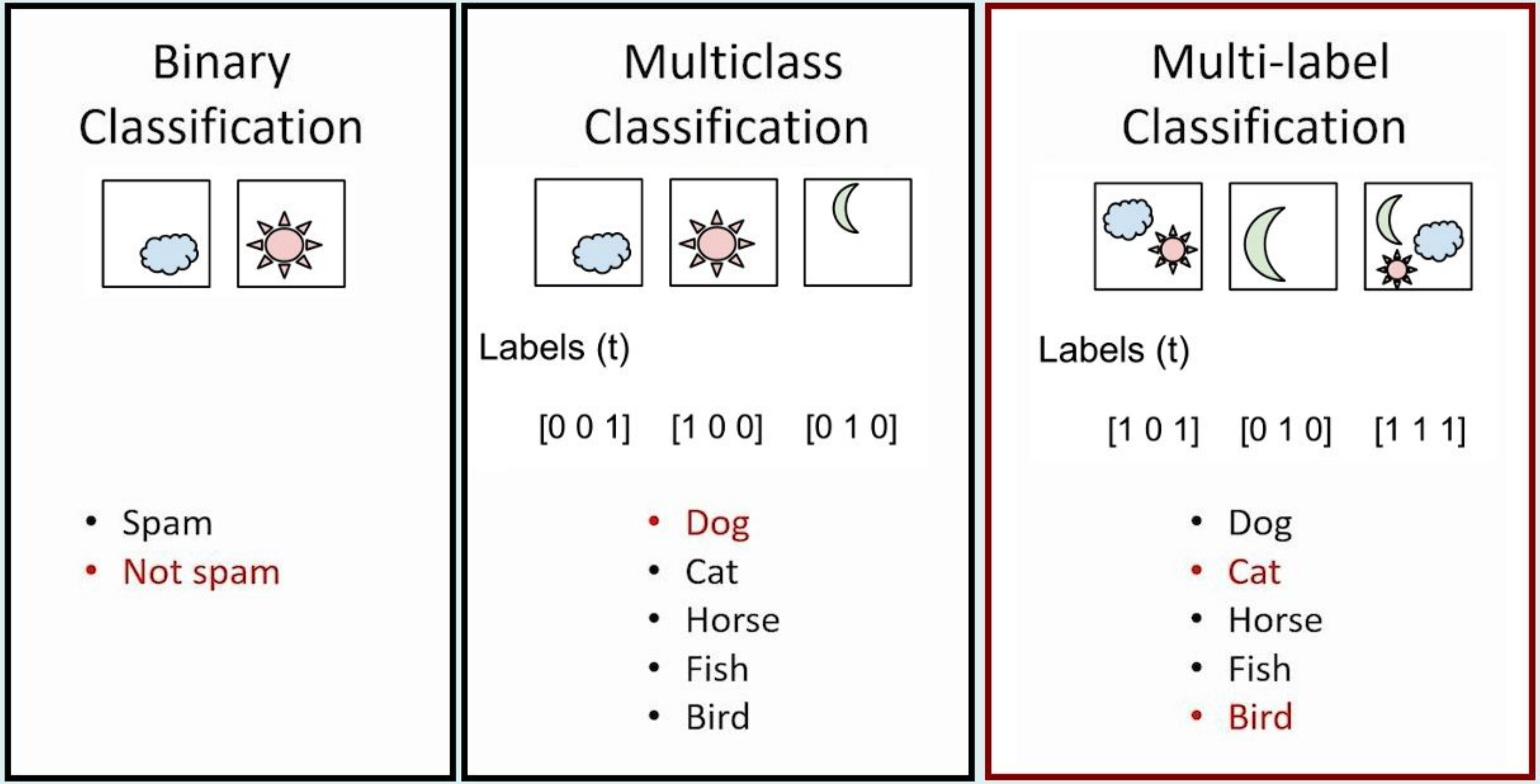
FEATURE RELEVANCE TO GENRE



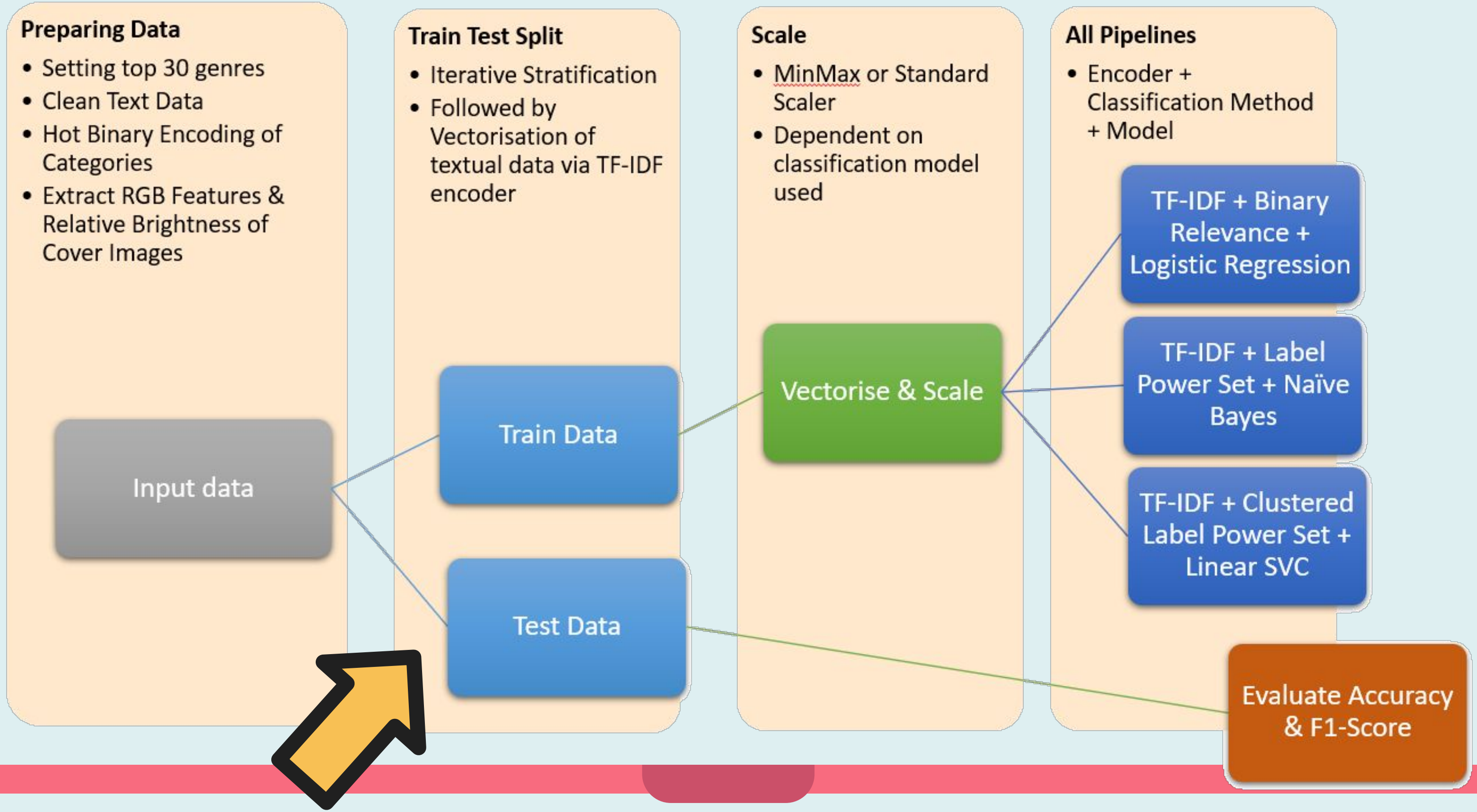


CLASSIFYING & MACHINE LEARNING

TYPES OF CLASSIFICATION

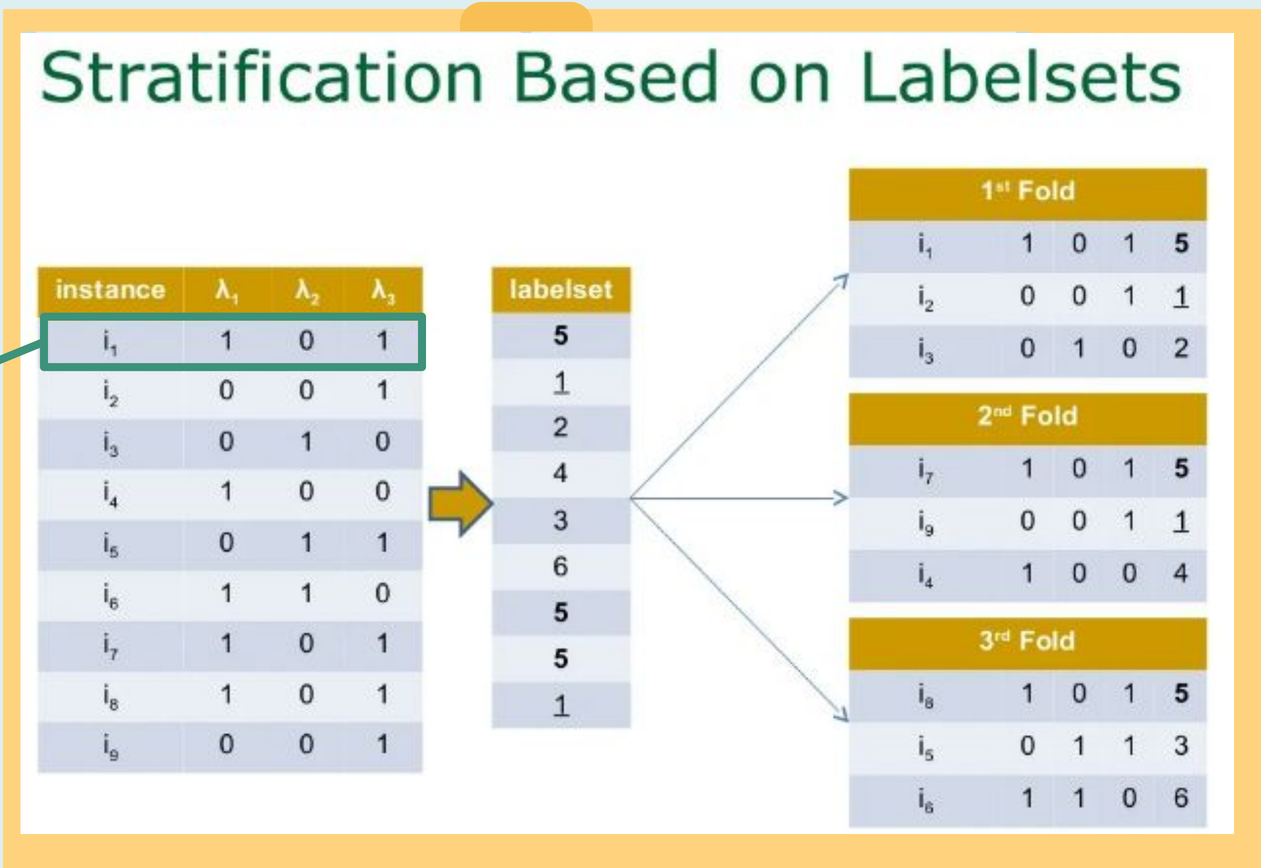


PROJECT PROCESS OVERVIEW



STRATIFICATION

One labelset



Stratification Based on Labelsets

instance	λ_1	λ_2	λ_3
i_1	1	0	1
i_2	0	0	1
i_3	0	1	0
i_4	1	0	0
i_5	0	1	1
i_6	1	1	0
i_7	1	0	1
i_8	1	0	1
i_9	0	0	1

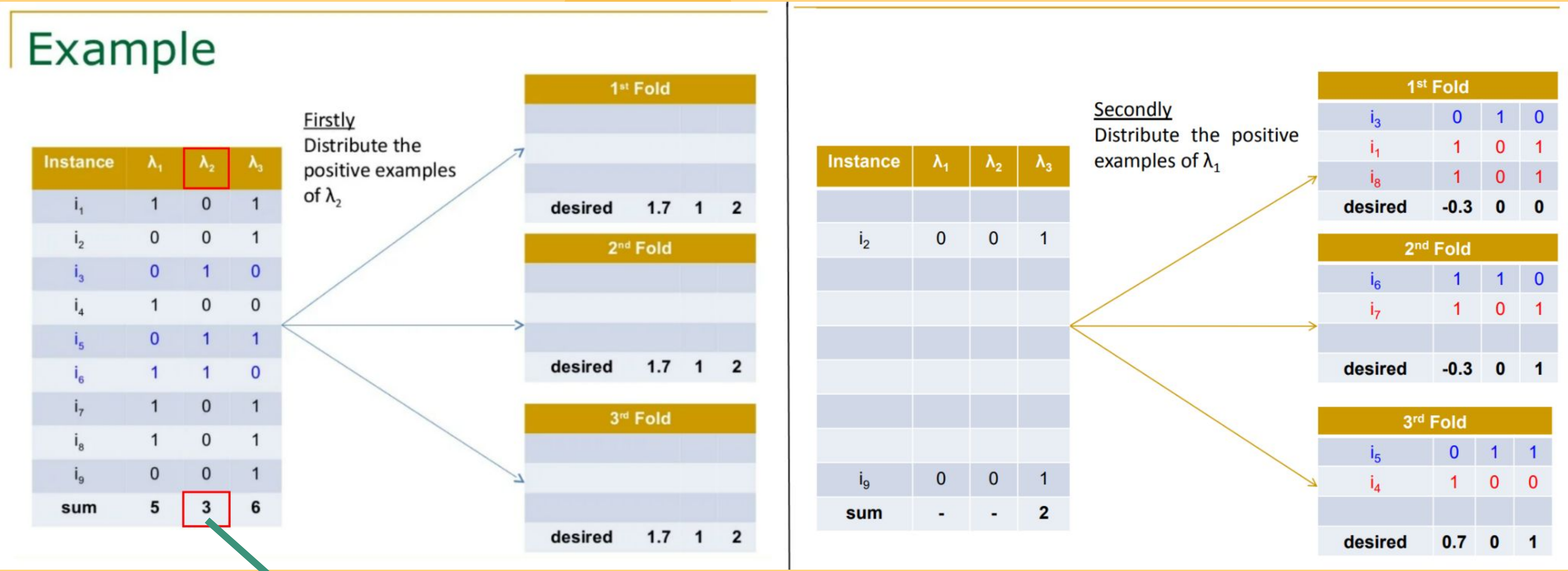
labelset
5
1
2
4
3
6
5
5
1

1st Fold				
i_1	1	0	1	**5**
i_2	0	0	1	1
i_3	0	1	0	2

2nd Fold				
i_7	1	0	1	**5**
i_9	0	0	1	1
i_4	1	0	0	4

3rd Fold				
i_8	1	0	1	**5**
i_5	0	1	1	3
i_6	1	1	0	6

STRATIFICATION



Example

Instance	λ_1	λ_2	λ_3
i_1	1	0	1
i_2	0	0	1
i_3	0	1	0
i_4	1	0	0
i_5	0	1	1
i_6	1	1	0
i_7	1	0	1
i_8	1	0	1
i_9	0	0	1
sum	5	$\boxed{3}$	6

Firstly
Distribute the positive examples of λ_2

1st Fold			
desired	1.7	1	2

2nd Fold			
desired	1.7	1	2

3rd Fold			
desired	1.7	1	2

Instance	λ_1	λ_2	λ_3
i_2	0	0	1
i_9	0	0	1
sum	-	-	2

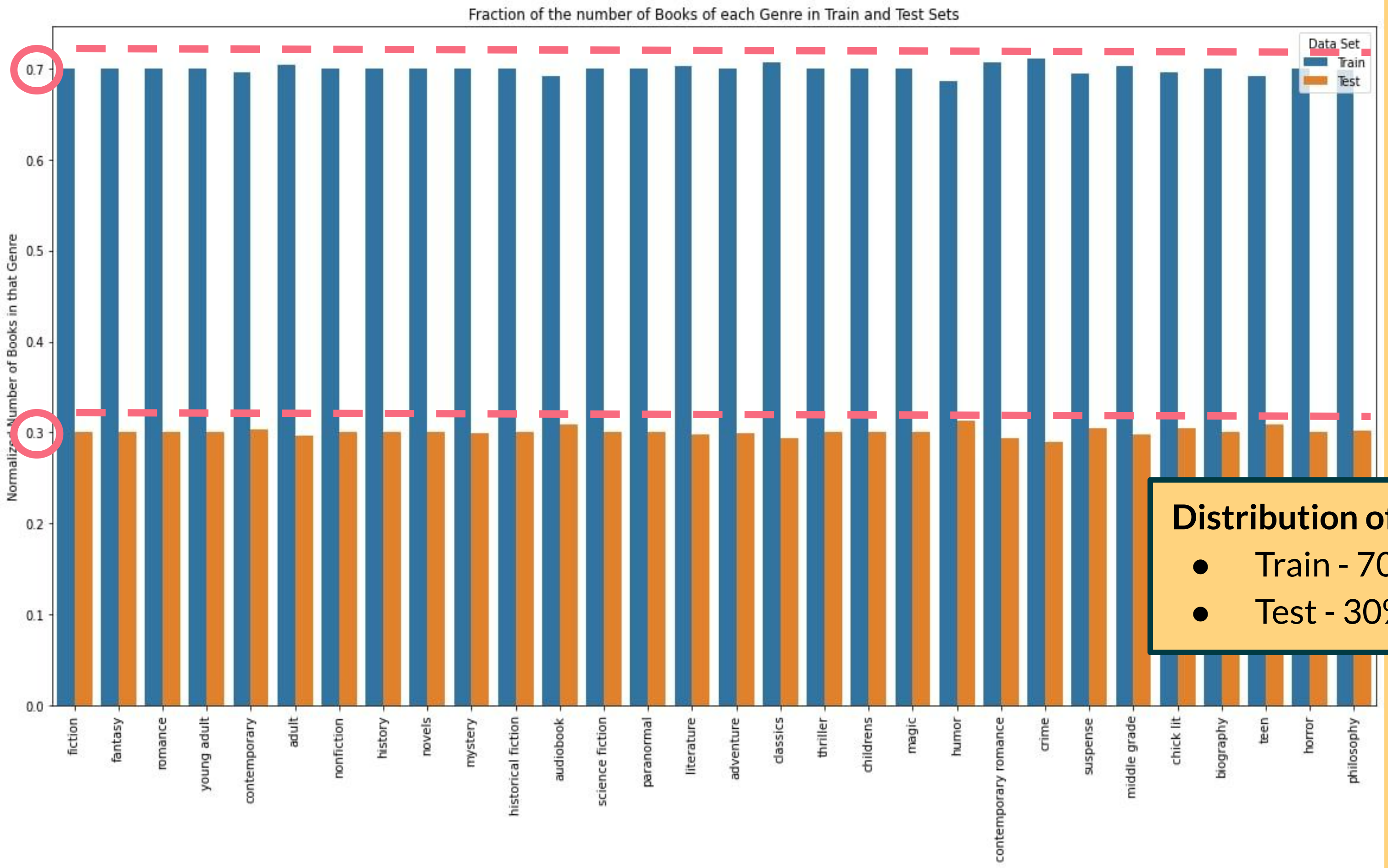
Secondly
Distribute the positive examples of λ_1

1st Fold			
i_3	0	1	0
i_1	1	0	1
i_8	1	0	1
desired	-0.3	0	0

2nd Fold			
i_6	1	1	0
i_7	1	0	1
desired	-0.3	0	1

3rd Fold			
i_5	0	1	1
i_4	1	0	0
desired	0.7	0	1

Λ2 is distributed first



Distribution of Train/Test Split

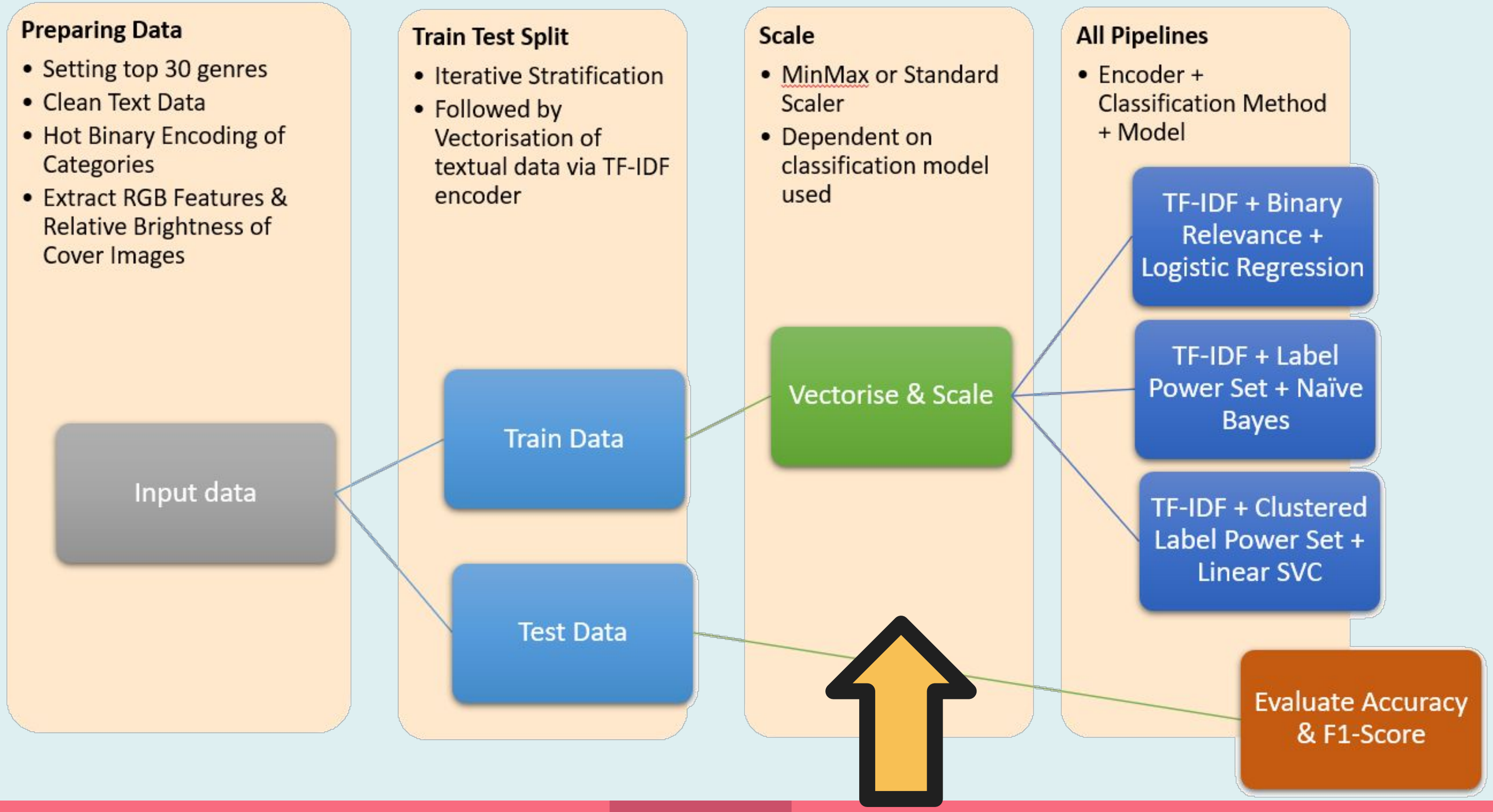
- Train - 70%
- Test - 30%

TRAIN/TEST SPLIT

Y	
Top 30 Genre	Standardised 30 Genres we decided on previously

X	
Brightness	Use of R,G,B values to calculate perceived luminance
Number of Ratings	Number of ratings the book has on GoodReads
Description	The book description given by the author

PROJECT PROCESS OVERVIEW



VECTORIZATION

Vectorization

- Similar concept to one hot encoding
- Convert text to numerical representation

$$IDF_i = \log\left(1 + \frac{N_D}{f_i}\right)$$

Inverse Document Frequency for the search term i within the corpus of documents

The number of documents in the corpus of documents that contain the term D

The number of documents that contain the search term

TF-IDF

- Term frequency-inverse document frequency
- Used to quantify the importance or relevance of string representation (words, phrases, etc)

MACHINE LEARNING

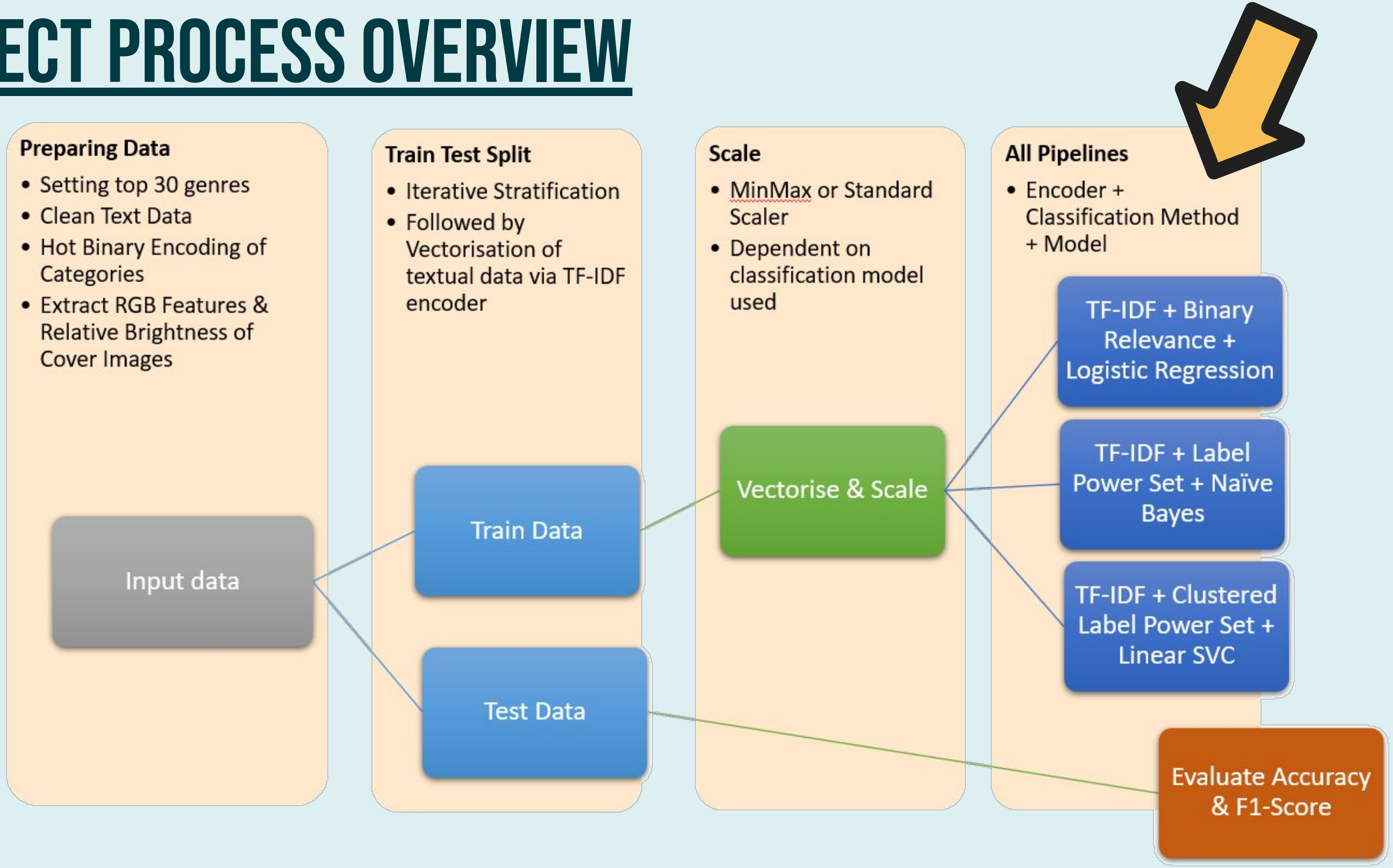
Scaling

- Our group decided to scale our data because ML algorithms are sensitive to data scales
- If the data is not scaled, the features with a higher value range starts dominating when calculating distances
- Our team choose **MinMax Scalar** because it scales from [0 to 1] which ensures that no negative values are returned when passing it into the different models

Machine Learning Algorithms

- Logistic Regression
- Naive Bayes
- Linear Support Vector Machine

PROJECT PROCESS OVERVIEW



MULTI-LABEL CLASSIFICATION ALGORITHMS

Binary Relevance

- Treat each label as a separate class classification
- 30 genres => 30 binary classifiers

X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Label Powerset

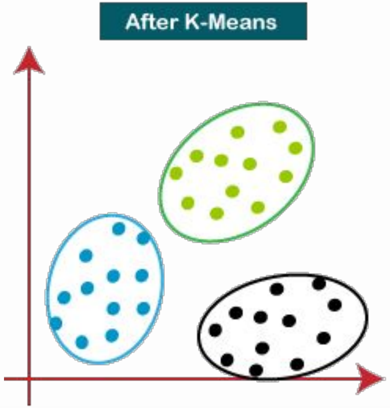
- Treat each unique genre combination as a class
- 6436 unique combinations => 6436 classes

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0
x4	0	1	1	0
x5	1	1	1	1
x6	0	1	0	0

X	y1
x1	1
x2	2
x3	3
x4	1
x5	4
x6	3

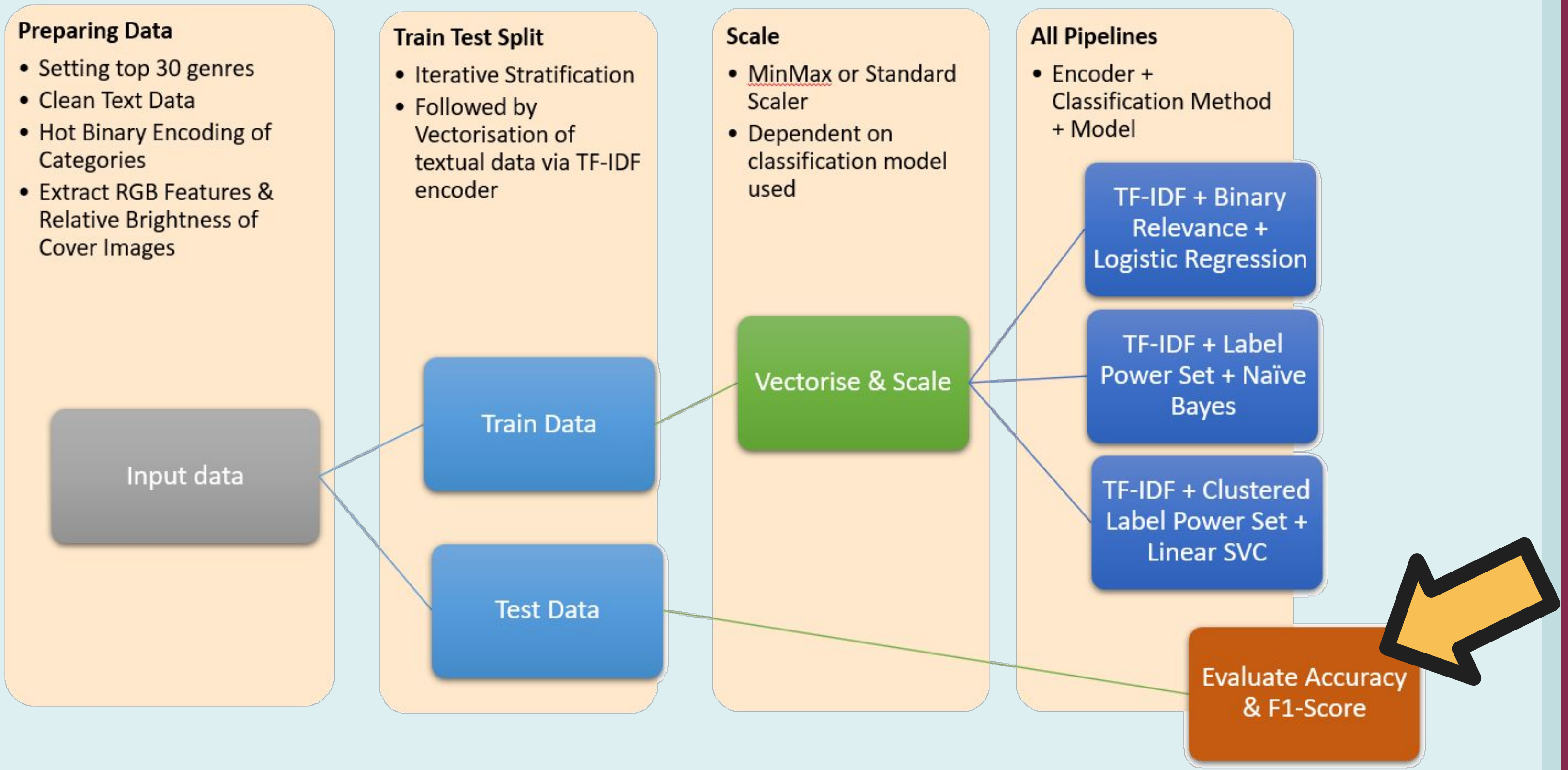
Label Powerset with Clustering

- Reduce the number of genre combinations by clustering from 6436 to 100
- k=100 (number of clusters) gave us the highest average F1-score

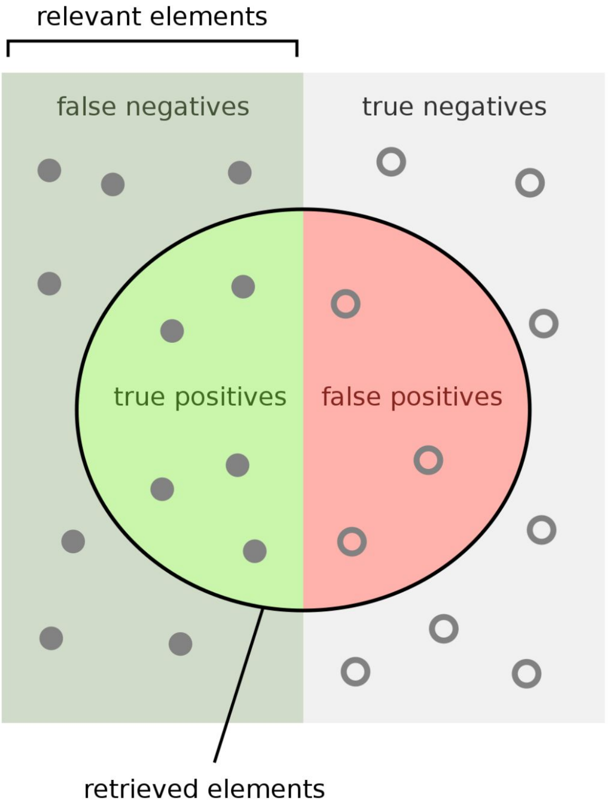


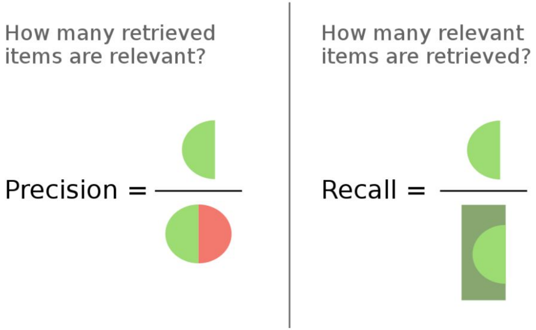
PIPELINE ANALYSIS

PROJECT PROCESS OVERVIEW



EVALUATION OF MODELS





$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}score = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

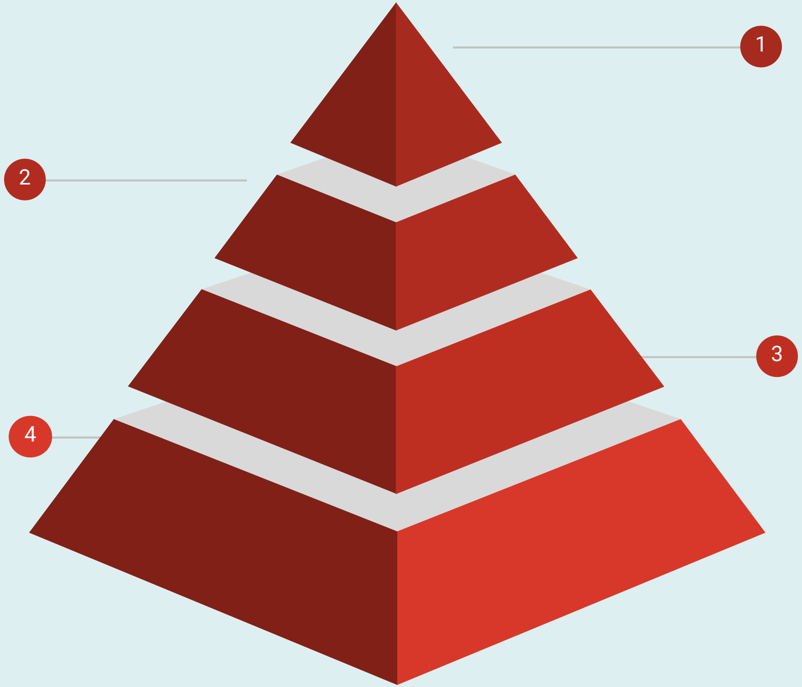
EVALUATION OF MODELS

	Precision	Recall	F1-Score	Support
fiction	0.82	0.62	0.71	6477.0
fantasy	0.83	0.42	0.55	3208.0
romance	0.72	0.55	0.62	3298.0
young adult	0.70	0.43	0.53	2774.0
contemporary	0.67	0.27	0.39	2157.0
adult	0.32	0.47	0.38	2039.0
nonfiction	0.47	0.79	0.59	2176.0
biography	0.20	0.59	0.30	681.0
teen	0.56	0.11	0.18	1034.0
horror	0.44	0.08	0.14	575.0
philosophy	0.64	0.31	0.42	475.0
Avg/Total	0.58	0.42	0.43	48132.0

EVALUATION OF MODELS

Machine Learning Model	Precision	Recall	F1-Score
Binary Relevance + Logistic Regression	0.65	0.26	0.29
Label Powerset + Naive Bayes	**0.58**	**0.42**	**0.43**
Label Powerset Clustering + Linear Support Vector Machine	0.29	0.26	0.27

FUTURE WORK



Expand the list of genres

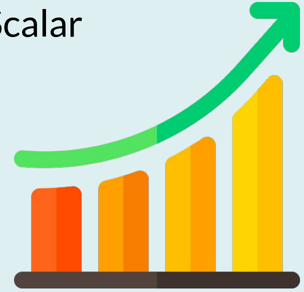
Increase from size 30 to cover a wider range of genres or make it more in depth

Utilise different Text Encoder

Google Universal Sentence Encoder

Utilising different Scaling techniques

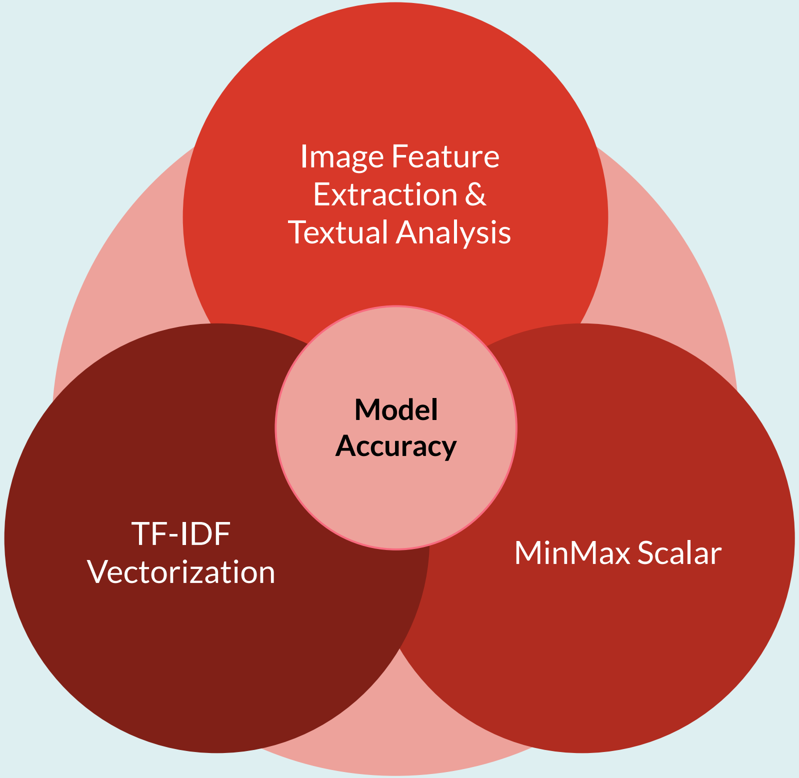
Robust Scalar



Utilising other Machine Learning Models

- Neural Network
- Cosine Similarity
- Adjusting hyperparameters

NEW TECHNOLOGIES USED

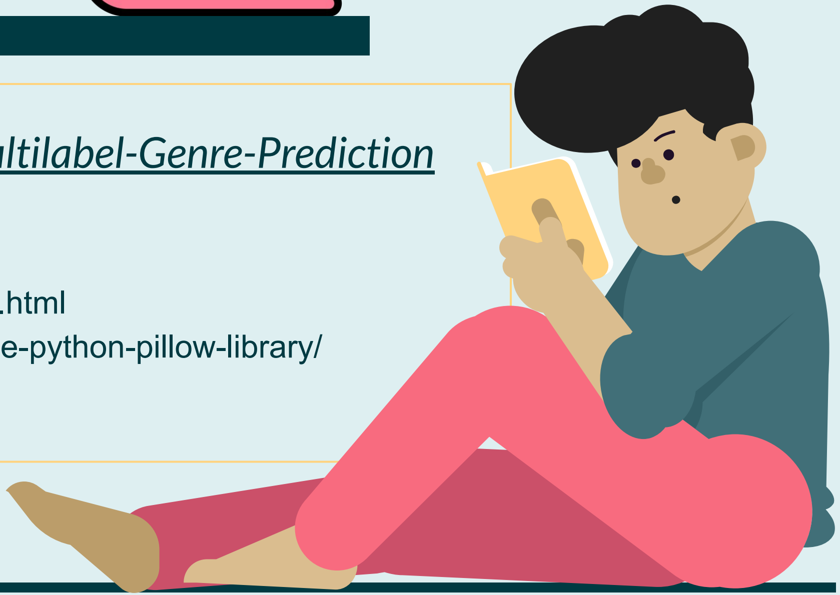


scikit learn

Train Test Split	**Iterative Stratification**
Binary Relevance	**Logistic Regression**
Label Powerset	**Naive Bayes**
Label Powerset with clustering	**Linear SVM**

THANK YOU!

https://github.com/natisaver/GoodReads-Multilabel-Genre-Prediction



References

- https://scikit-learn.org/stable/modules/multiclass.html
- https://realpython.com/image-processing-with-the-python-pillow-library/