Capstone Project

BOOK REVIEW PREDICTOR

With supervised classification and unsupervised clustering

Group 1:

Vũ Công Duy - 20176737

Trần Công Minh - 20176825

Đào Hồng Quân - 20176850

CONTENTS:

I. Introduction

1. Project overview

The project involves a classification model that predicts whether a book has negative, mixed, or positive reviews from the public (goodreads.com).

The main attributes for classification include the number of readers (raters), the number of comments (reviewers), the book's length (pages), and its original publication year.

Additionally, a book also has "tags", which is the genres given by the readers. These non-numeric values can be used to enhanced the efficiency of the classifier.

The book data will be clustered into groups and classified from each cluster for hopefully better prediction result.

2. Machine learning methods

2.1 Unsupervised learning

The books will be separated into groups by a variation of the clustering method K-Means called K-Modes.

2.2 Supervised learning

The methods for review classification include:

- K-Nearest Neighbors
- Gaussian Naïve Bayes
- Decision Tree

3. Variables

3.1 Unsupervised learning

The exploratory variables of this model are the list of authors, the book's series (if exists), and a list of its tags. All of these are represented in string.

3.2 Supervised learning

Classification methods rely on four fields:

- The number of raters
- The number of reviewers
- The number of pages
- The publication year

The response variables are: 0 for “negative”, 1 for “mixed”, 2 for “positive”.

II. Data

1. Data scraping

500,000 books for this project are crawled from website “goodreads.com” and saved in csv files.

Most of the books’ data can be found on its page with an url containing its ID. The tags of the books are crawled from the link on its page.

Scrapy is used for this task. Any sites with errors can be ignored thank to the large quantity of the data.

2. Data cleaning

The data are cleaned with the pandas and numpy libraries on Jupyter Notebook.

Books without tags, unpopular books (less than 100 raters), books without any page (audio book), books with faulty publication year are considered noises and removed from the data.

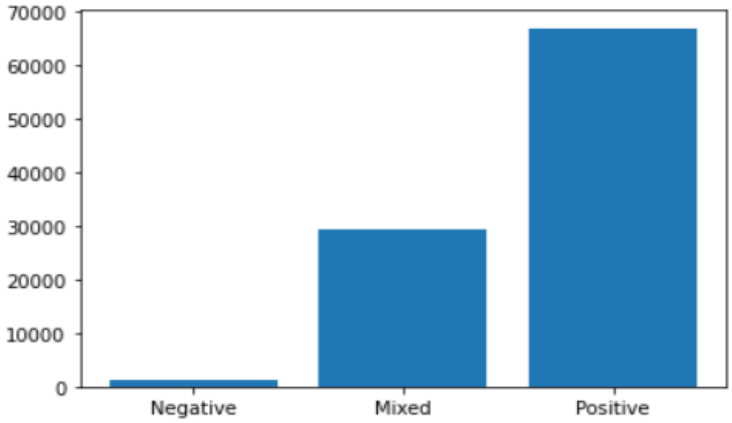
From the original 456,000 – row dataset, only over 97,000 books remain.

3. Additional processes

Since ‘goodreads.com’ only has float values for rating, the books will be classified into:

- 0: less than 3.2 rating.
- 1: from 3.2 to 3.8 rating
- 2: above 3.8 rating

However, there is a great unbalance in the number of books of each type, which will cause issues in some classification models.



The original data crawled for tags are stored in a normalized csv table. Tags of the same books are concatenated into a single string, which greatly reduces the size.

Both the main book table and tags table are joined and stored in a main data frame for the project.

	BookID	Title	Author	Rate	Raters	Reviewers	Pages	PublishYear	GenreLink	Series	review	Tags
0	1	Harry Potter and the Half-Blood Prince	J.K. Rowling	4.57	2469197	40043	652.0	2005	/work/shelves/41335427	Harry Potter	2	to-read, fantasy, favorites, young-adult, fict...
1	2	Harry Potter and the Order of the Phoenix	J.K. Rowling, Mary GrandPré	4.50	2541611	43067	870.0	2003	/work/shelves/2809203	Harry Potter	2	to-read, currently-reading, fantasy, favorites...
2	3	Harry Potter and the Sorcerer's Stone	J.K. Rowling, Mary GrandPré	4.47	7145889	113905	309.0	1997	/work/shelves/4640799	Harry Potter	2	to-read, currently-reading, fantasy, favorites...
3	4	Harry Potter and the Chamber of Secrets	J.K. Rowling	4.43	2766218	53651	352.0	1998	/work/shelves/6231171	Harry Potter	2	to-read, currently-reading, fantasy, favorites...
4	5	Harry Potter and the Prisoner of Azkaban	J.K. Rowling, Mary GrandPré	4.57	2849671	56337	435.0	1999	/work/shelves/2402163	Harry Potter	2	to-read, fantasy, favorites, currently-reading...

For testing, the data is also separated into clustering and classification csv files with their respective attributes.

III. Unsupervised clustering

1. Problem statement

A book has not only its official genre but is also frequently branded other tags by online readers, such as “favorite” or “to-read”. There can be a hidden pattern in books with similar tags. The authors and the book’s series are also factors in the readers’ trend.

However, K-Means clustering only works for numerical attributes. Therefore, a different clustering method is required.

2. K-Modes

2.1 Introduction

K-Modes is a modified version of K-Means where the centroid is determined by the mode of each attribute of the data rather than the means.

2.2 Algorithm

2.2.1 Distance measurement

Since the data fields are in string type, they will be measured in differences by a variation of the hamming distance:

- Each string is separated into groups of words (author’s name or individual tag).
- The original distance between two strings is the length of the longest list.
- For each element in the first list, if it appears in the other lists, the distance is decreased by 1.
- The result will be normalized by dividing with the greater length of the two lists.
- The distance of all string attributes is summed up and returned as the distance between two books.

By this way, the difference between books will not be too big and the complexity of the algorithm is better than only comparing the difference in each string.

2.2.2 Mode calculation

In each cluster, there will be a dictionary of frequency for authors, series, tags, and the number of tags for each book.

The author and series with the highest frequency will be selected. The most common length determines the number of tags chosen by descending frequency order.

```
[['Neil Gaiman',
  'none',
  'to-read, fantasy, comics, graphic-novels, neil-gaiman, fiction, graphic-novel, own, audio_wanted, vertigo, favorites, owned, sandman, comics
-graphic-novels, audio-wanted, books-i-own, comic-books, reference, gaiman, horror, non-fiction, illustrated, currently-reading, graphic-novels
-to-read, fantasy-sci-fi, quotes, wishlist, to_read, pop-culture, to-look-for, english, to-read-scifi, other, comic, fiction-fantasy, comics-an
d-graphic-novels, sff, owned-books, default, sci-fi, series, mythology, hardcover, to-buy, male-author, graphic-novels-comics, gods, boekenkas
t, wanted, speculative-fic, highly-recommended, allison-s-books, coffee, writing, art, _location-basement, e-books, speculative-fiction, comics
-manga-graphic-novels, don-t-own-yet, ll-want_audio, ll-read, current_test, want-to-read, have, manga-graphic-novels, my-books, comics-manga, r
ead-selected, four-star, my-collection, fantasy-sci-fi-horror, owned-books-all, owned-nonfiction, bpl, my-real-bookshelf, comics-graphic-novels
-bart, art-books, comic-graphic-novel, want-to-read-again, graphic-novels-cartoons-and-comics, fiction-graphic-novels, known-authurs, fiction-g
-l, non-czech, in-my-library, fantasy-scifi, fantasy-science-fiction, no, library-to-read, gaiman-to-read'],
```

2.2.3 Algorithm

Step 1: K points are chosen randomly from the dataset and used as the original centroids.

Step 2: For each point in the dataset, the distance to each centroid is calculated. The point is assigned to the cluster of the nearest centroid. The frequency dictionary of the cluster is updated with the new point.

Step 3: The mode of each cluster is calculated based on its frequency dictionary. For each point in the cluster, assign the one nearest to the mode as the new centroid.

Step 4: Calculate the distance between each cluster's old and new centroids. Repeat from step 2 until all distances become zero.

2.2.4 Inertia

To determine the compactness of a clustering model, the inertia is calculated by adding up the distances of all data points to their respective centroid.

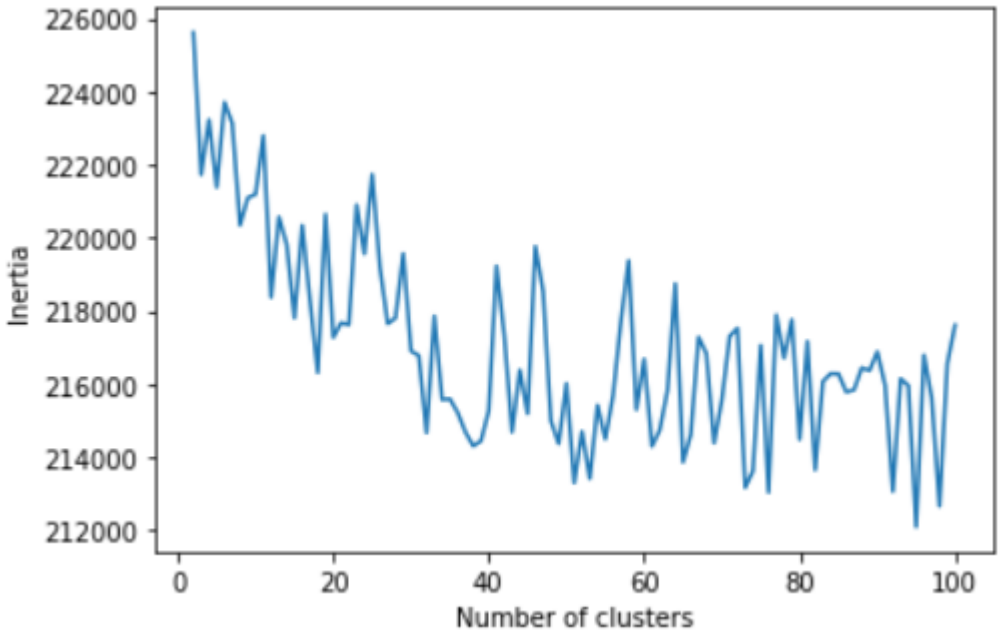
3. Implementation

The model is coded as a class object with the aim to reuse in future project. Therefore, the functions mentioned beside the algorithm are passed as arguments rather than actually implemented in the class.

4. Results

Since the algorithm deals with a lot of string operations, the runtime rises greatly with the increase of data points or number of clusters.

It took 14 hours to determine the inertias of models ranging from 2 to 100 clusters.



The result fluctuates greatly, but slowly declines. As we can see, 30 is the most suitable number of clusters as the inertia rate overall reduces drop rate afterward.

IV. Supervised classification methods

1. Problem statement

The main goal of project is to determine a book's general opinion based on its existing information. The public view is very biased, which can be a challenge for prediction.

The efficiencies of three basic classification methods are examined.

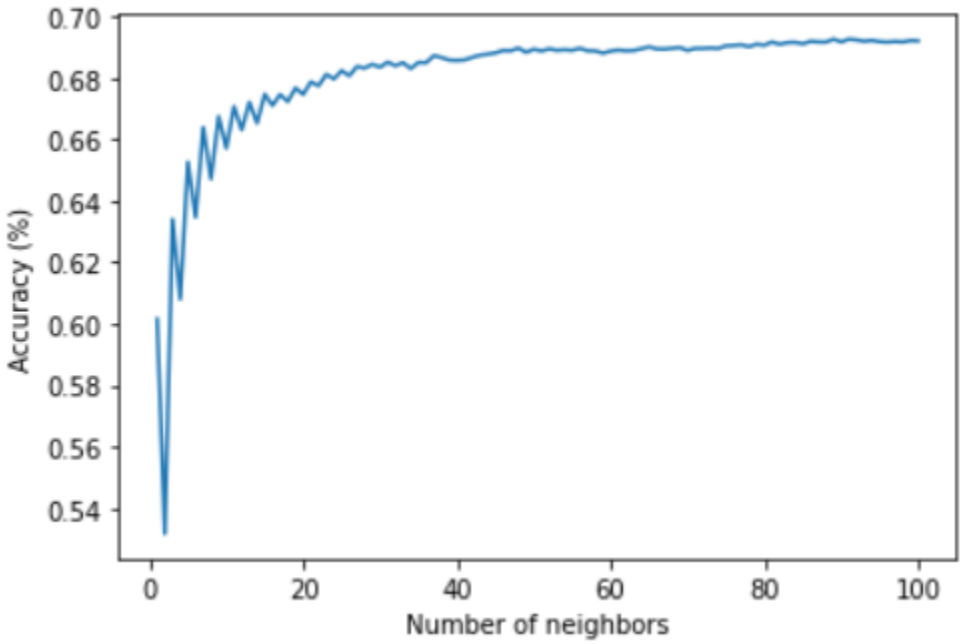
2. K-Nearest Neighbors

2.1 Implementation

In this project, we use the KNeighborsClassifier model from the scikit-learn library.

2.2 Results

The accuracy of this model is between 54% and 70%.



As the number of neighbors reaches 40, the model improves less significantly. Therefore, the suitable number of neighbors should be between 20 and 40.

However, since the number of negative books is too small, high number of neighbors will ignore the “negative” predictions.

```
[   13,   128,   133]
[  180,  1796,  3873]
[  203,  2627, 10568]
```

K = 3

```
[    0,   109,   165]
[    0,   975,  4874]
[    0,   956, 12442]
```

K = 37

In the confusion matrices above, the column indexes represent the number of labels predicted and the row indexes represent the number of actual results. The diagonal line represents the correct results.

3. Naïve Bayes

3.1 Implementation

Since the data are numeric, the Gaussian model GaussianNB of the sklearn.naive_bayes library is used.

3.2 Results

The model has extremely poor result of around 15% correct. However, the model has a fairly distributed prediction.

```
[ 241,   26,    7]
[4492, 1019,  338]
[8279, 3324, 1795]
```

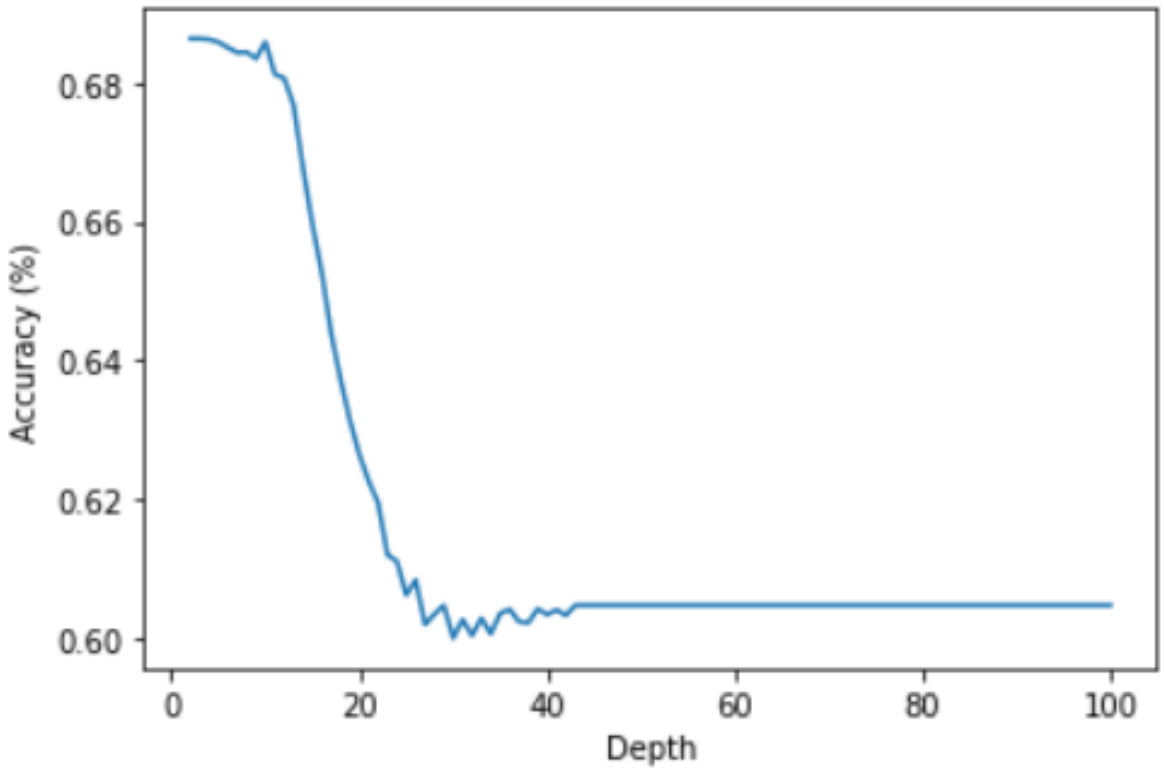
4. Decision Tree

4.1 Implementation

We use the DecisionTreeClassifier model from sklearn library.

4.2 Results

At low depth, the model has an accuracy of around 68% but greatly reduces with the increase in depth.



However, even at its best result, the model suffers great bias as it usually returns the result of most frequent label.

```
[    0,     0,   274],
[    0,     0,  5849],
[    0,     2, 13396]]
```

5. Comparison

The Naïve Bayes model gives the worst result but has the most spread-out predictions.

While the decision tree has the same accuracy as KNN's, it is a result of bias rather than efficiency.

KNN seems to be the most adaptive model with constantly high accuracy. However, the model takes the most time to run.

V. Book review predictor

1. Overview

The program is combination of the K-Modes model and a classification method of choice.

2. Implementation

The model is coded as the class containing a K-Modes model. The functions mentioned in the K-Modes section are also used here.

When a dataset is used to train the model, it is divided into clustering data for the K-Modes model and classification data for other classifiers to use. After clustering, a list of clusters is created. Each classification data point and its label is appended to its respective cluster.

When a classifier is passed into the model, each cluster will generate its own classifier from the original one and train it with the cluster data.

When the model's predict method is called, it receives the dataset and a classifier as arguments. Each data point is predicted by the classifier of its respective cluster.

3. Results

At 3 clusters, overall, there is only slight improvement in comparison with the sole classifier. In Naïve Bayes case, the accuracy increases by over 10%.

With the ideal number of clusters as 30, the book model can reach up to 70% correct with the help of a good KNN model. The Naïve Bayes method can also reach up to 40%.

In all cases, K-Modes seems to have little impact on the accuracy of Decision Tree, but the bias in prediction is reduced.

However, in some cases, the number of elements in a cluster can be smaller than the number of neighbors required for KNN, which may cause error. This also raises the issue of underfitting in small clusters.

The runtime will also increase with more complex classifiers.

VI. Overview

1. Data

While a large number of data is crawled, most of them are noises and cannot be used. The data itself is also severely affected by users' bias. Therefore, the accuracy can hardly get over 70%.

2. K-Modes Algorithm

This model is still very basic and has high runtime complexity, which is a major drawback.

This clustering method is best used to boost the performance of probabilistic classification methods. As for KNN and decision trees, the runtime may cause more harm than good.

VII. Member contribution

Vũ Công Duy: K-Modes and notebooks

Đào Hồng Quân: K-Nearest Neighbors and data crawling

Trần Công Minh: Naive Bayes, Decision Tree, and data cleaning