EGR 598: Machine Learning and Artificial Intelligence – Final Project

Predict the rating for a User / Item pair using Jaccard Similarity

Team Members : Abhishek Kemia, Aman Mehul Patel, Ezhilan Veluchami, Kirthik Roshan Nagaraj, Shiva Sam Kumar

Arizona State University

Problem Statement

The aim of rating prediction is to determine how each user will rate the new books they read in the future.

This can be achieved by various Machine Learning Algorithms like Linear Regression, Logistic Regression, Bag of words and Jaccard Similarity. So, the problem that is going to be investigated in the project is as follow:

Which machine learning approach performs better in terms of accuracy and Mean Square Error to predict the user rating based on his/her rating and reviews for the previous books?

Data Description

Attribute Information Of Good Reads Datasets :

Key	Item Description
user_id	A unique user_id given to every new reader.
book_id	A unique book_id for which is picked up by the user.
review_id	ID generated when a user submits a review and rating.
rating	Rating of the book given by the user.
review_text	Given review by the user for the given book

Data Description

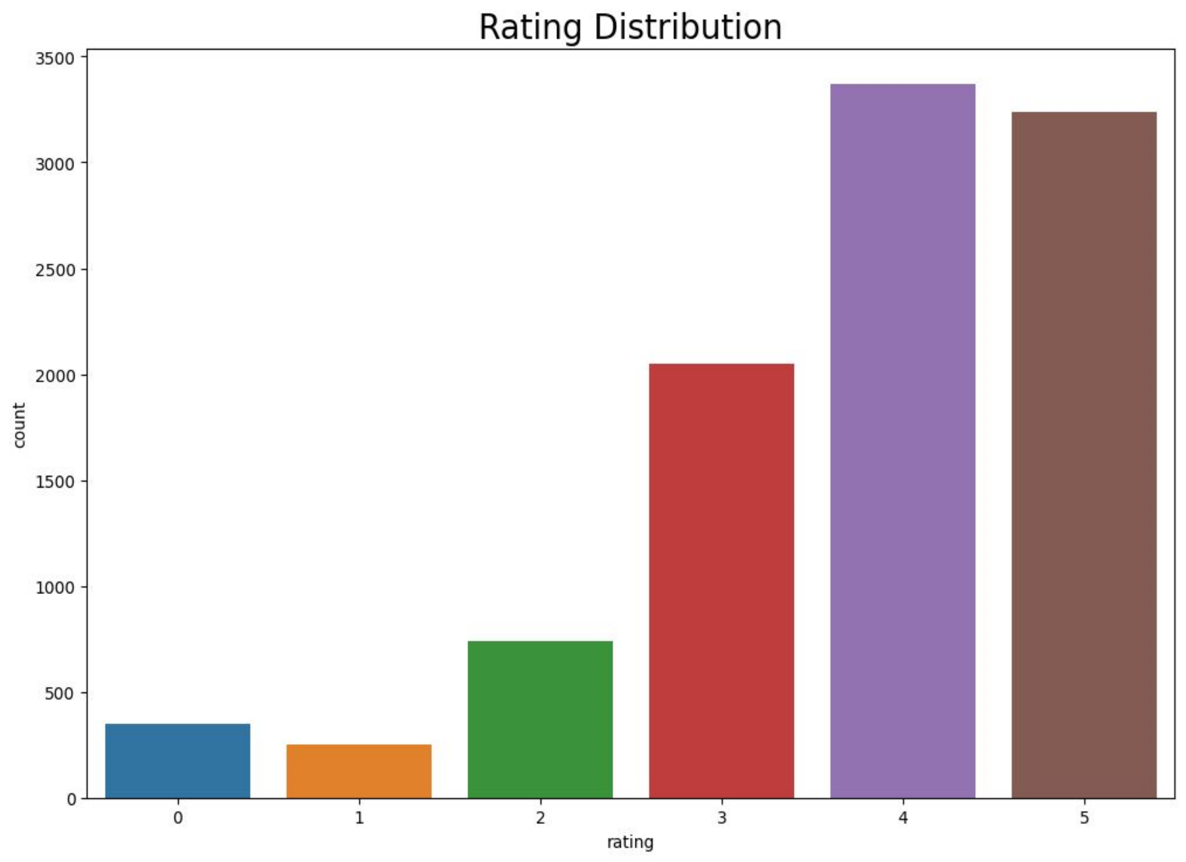
Key	Item Description
date_added	Gives the date the book was added in the dataset.
data_updated	Gives the date the user updated the book in the dataset.
read_at	Gives the date user started reading the book.
n_votes	Number of votes given to the user for the given rating and review.
n_comments	Number for comments given to the user for the given rating and review.

Data Preprocessing and Visualization

Preprocessing

- Checked data for null/empty values
- Encoding user_id and book_id field form hashed string to integers
- Removing primary key “review_id” as it is a unique value for all data points and have no correlation with the rating
- Removed the time stamps fields "date_added","date_updated","read_at" and "started_at" as they do not add any value to the Model
- Removed fields "n_votes" and "n_comments" as they were null/empty for more than 95% of all data points

Visualization



Initial Approach

1 Loading the Data set and Preprocessing

2 Attempting “Bag of words” approach to predict the user Rating by using Linear and Logistic regression

3 Defining New algorithm based on Jaccard similarity

4 Model Evaluation

Traditional Method

- A regression model provides a function that describes the relationship between one or more independent variables and a response, dependent, or target variable.

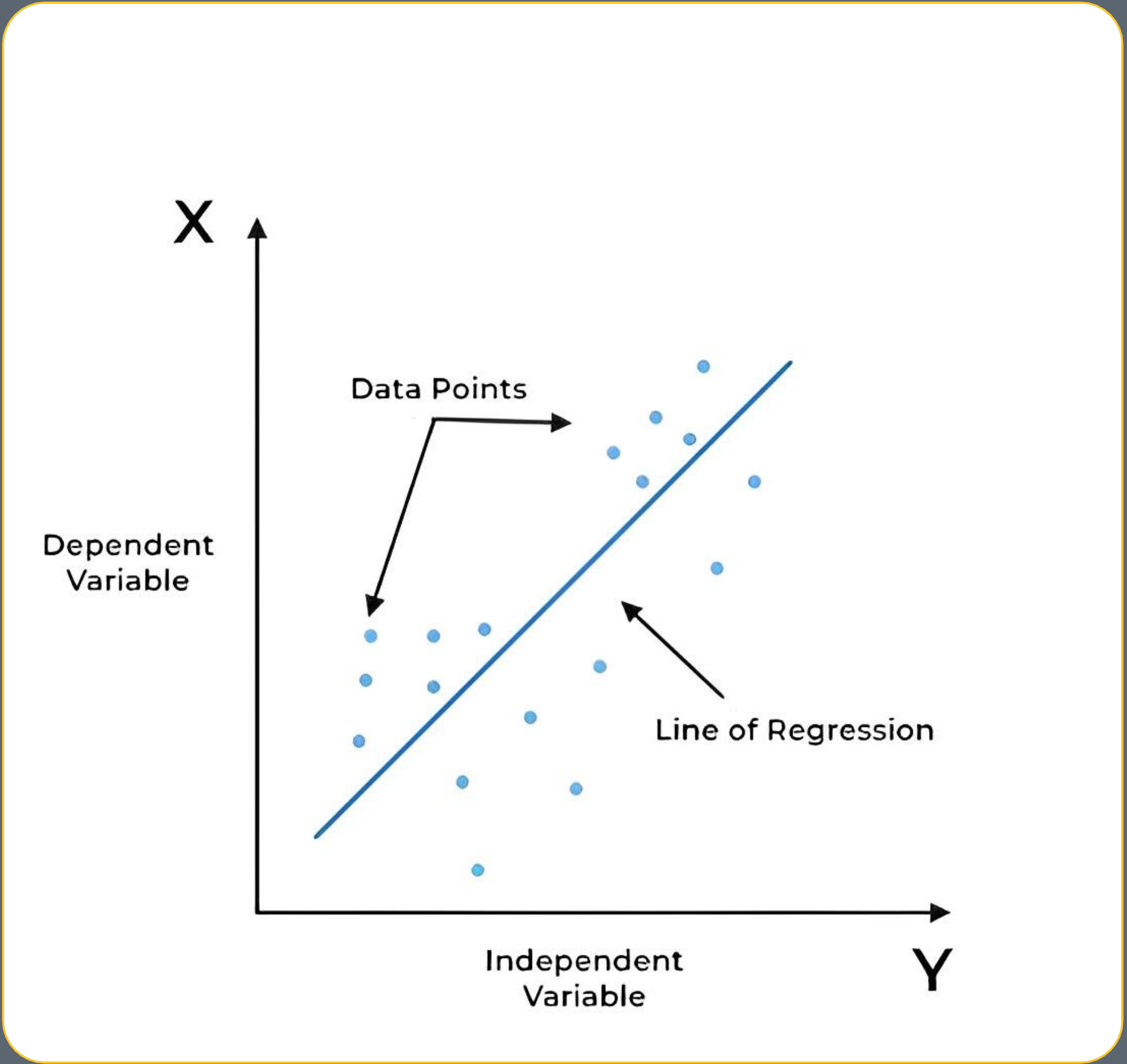
- A regression analysis is the basis for many types of prediction and for determining the effects on target variables.

- When you hear about studies on the news that talk about fuel efficiency, or the cause of pollution, or the effects of screen time on learning, there is often a regression model being used to support their claims.

Example

Linear Regression

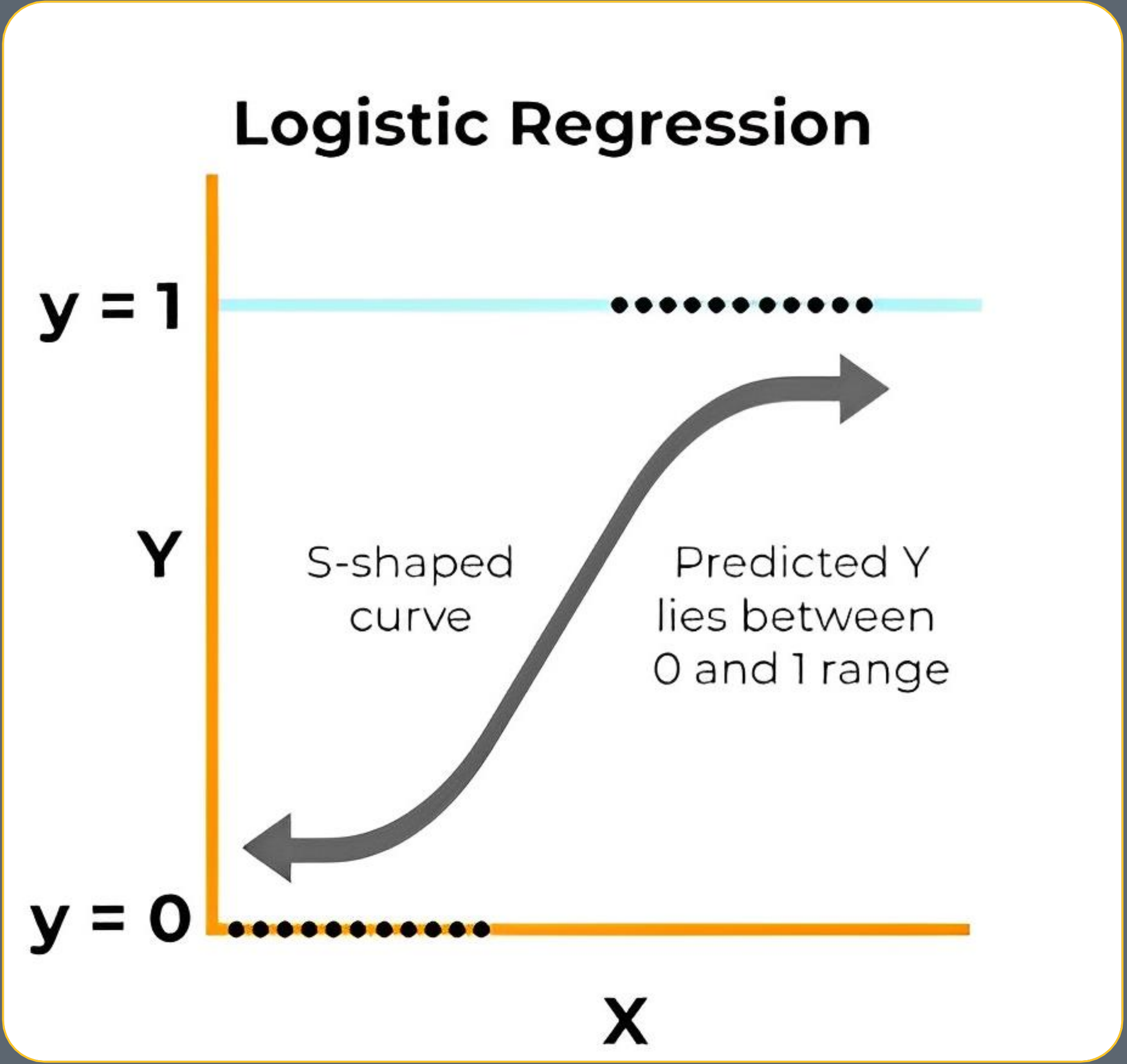
- Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.



Example

Logistic Regression

- Logistic regression is a classification algorithm. More specifically it’s a binary classification problem. Using nonlinearity, logistic regression classifies data points into classes. It uses a logistic function to learn weights to classify data points in classes, it resembles an “S” shaped curve which acts as the boundary of two classes.

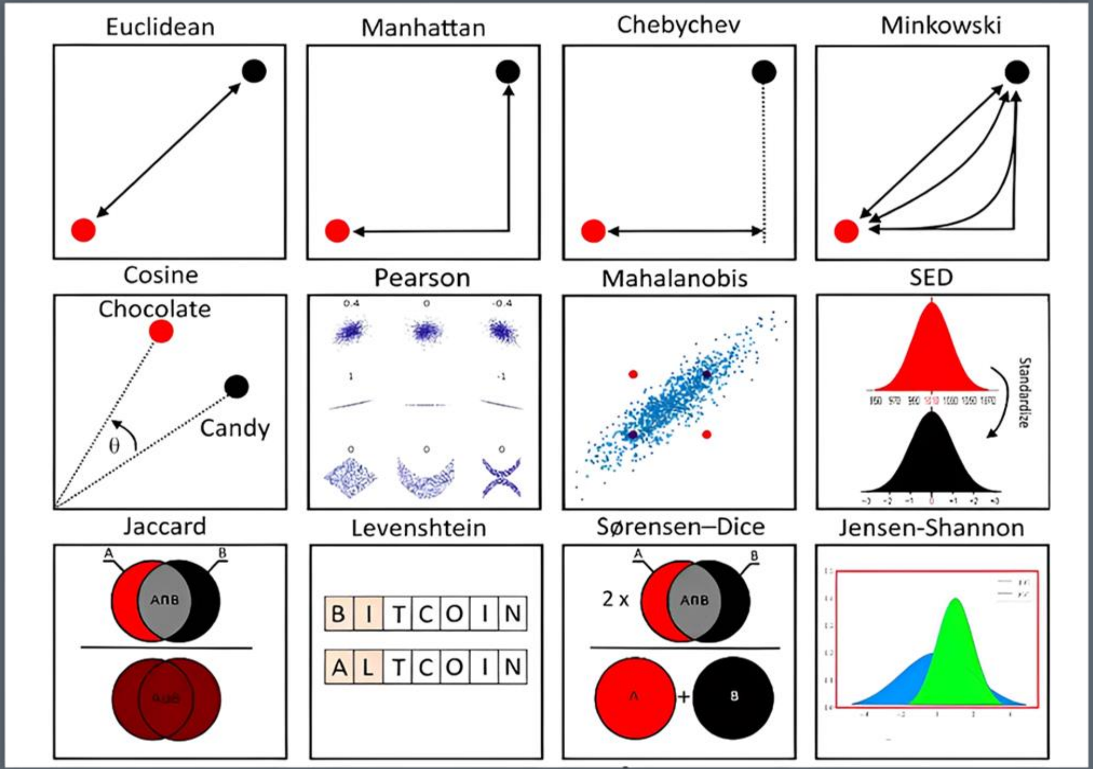


Similarity Index

- The similarity measure is a function that defines the similarity between two objects.
- The similarity measure is often presented as a numerical value, and it increases as the similarity between the data samples as it increases.
- It is often expressed as a number between zero and one by conversion: zero means low similarity, and one means high similarity.
- The smaller the distance is, the larger the similarity will get.

Similarity Index

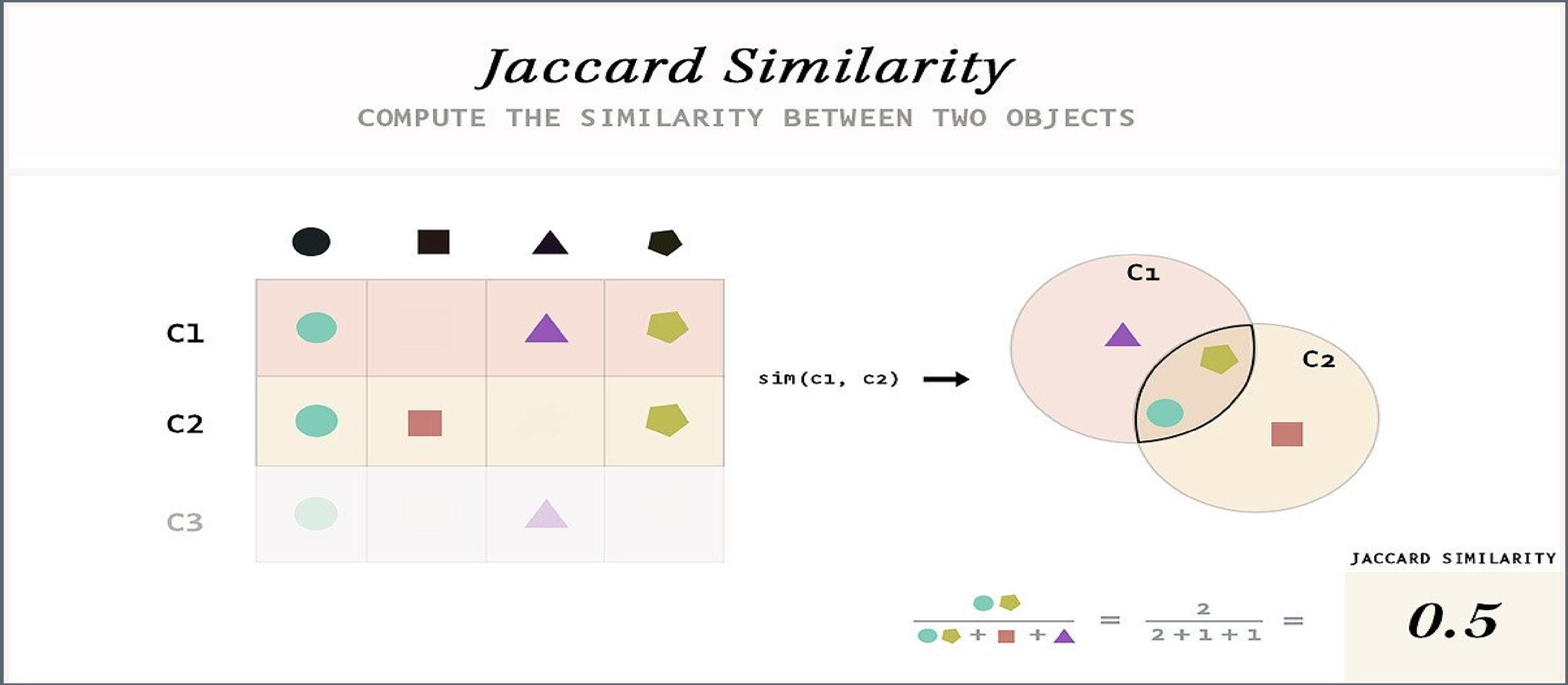
- The given similarity is Metric , if and on if ??
- Non -Negativity
- Symmetry
- Triangle Inequality



Jaccard Similarity Index

- Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents.
- Jaccard similarity can be used to find the similarity between two asymmetric binary vectors or to find the similarity between two sets.
- The Jaccard similarity measures the similarity between two sets of data to see which members are shared and distinct.
- The Jaccard similarity is calculated by dividing the number of observations in both sets by the number of observations in either set.

Jaccard Similarity Index



Jaccard Similarity

COMPUTE THE SIMILARITY BETWEEN TWO OBJECTS

	●	■	▲	⬟
C1	●		▲	⬟
C2	●	■		⬟
C3	●		▲	

sim(c1, c2) →

JACCARD SIMILARITY

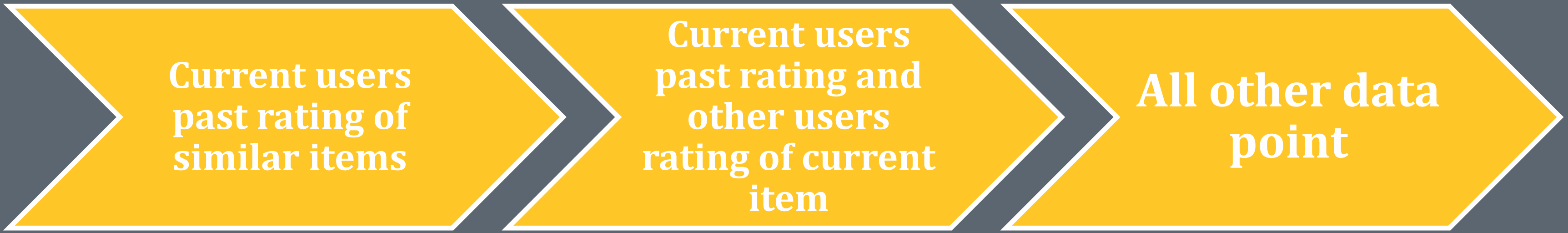
$$\frac{\bullet\ ⬟}{\bullet\ ⬟ + ■ + ▲} = \frac{2}{2+1+1} = 0.5$$

Ratings Based on Similarity Index

- When Predicting rating of a User/Item pair using traditional methods:
  - All the data points will have equal weights
  - Rating given by a different user for a different item will have same weight as rating given by the current user to a similar item
  - Ratings are predicted the with respect to the High frequency words
  - These weight should be modified to achieve a better model

Ratings Based on Similarity Index

- While Predicting Ratings based on Similarity the weight are given in the following:



Current users past rating of similar items

Current users past rating and other users rating of current item

All other data point

- This is because the most similar items should be the most relevant when predicting future ratings

Benefits of Jaccard Similarity

- Using Similarity based measurements for rating prediction approach, the rating given by a user can be estimated from the ratings that user has given to similar items.
- Jaccard Similarity is a weight-based approach which gives the user’s past ratings the highest weights. This helps in finding the most relevant future ratings.
- Traditional model predicts the data with respect to the High frequency words (The same words can be also corresponding to the attribute with a lower rating).

Bag of Words

The bag-of-words model is a condensing representation used in information retrieval and natural language processing (NLP).

A bag-of-words is a textual illustration that shows where words appear in a manuscript. There are two components:

- collection of well-known words.
- metric for the number of well-known words.

Bag of Words

Steps Followed :

- Collection of String Data
- Selection of useful words
- Creating Words Matrix
- Managing Vocabulary
- Scoring Words

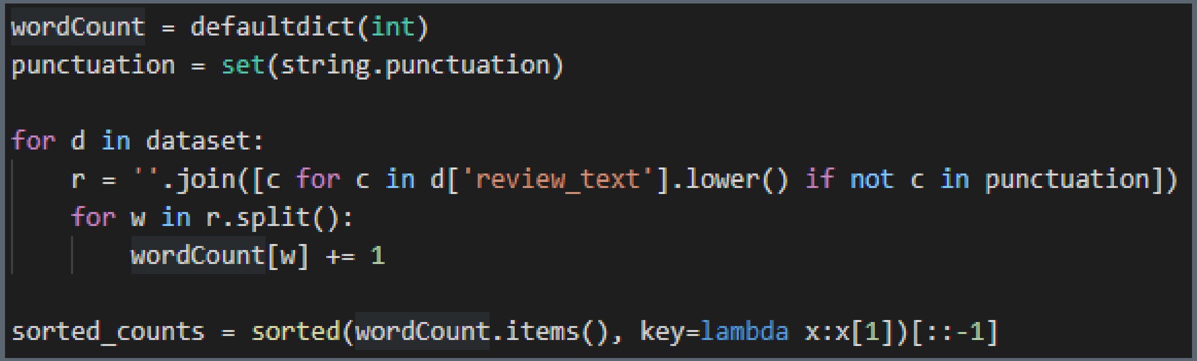
Bag of Words

Limitations !!

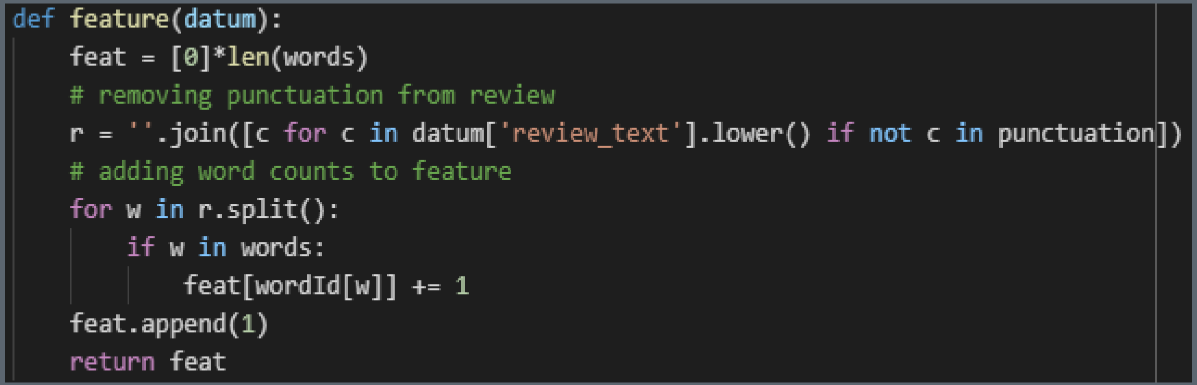
- Vocabulary: The vocabulary needs careful design, especially to control the size, which affects how sparsely it is represented in the page.
- Sparsity: Sparse representations are more difficult to model for computational and informational reasons.
- Meaning: By disregarding word order, the context and subsequent meaning of the document's words are disregarded . The model could recognize the difference between the identical words differently organized.

Code Walk Through

- Code Variables for Bag of Words Approach :
  - Word Count - Keeps count of repeated unique words in the dataset.
  - words - Stores the most common occurring and frequently used words in the dataset.
  - function <feature> - Generates a list of the word's matrix with the used words per data input.



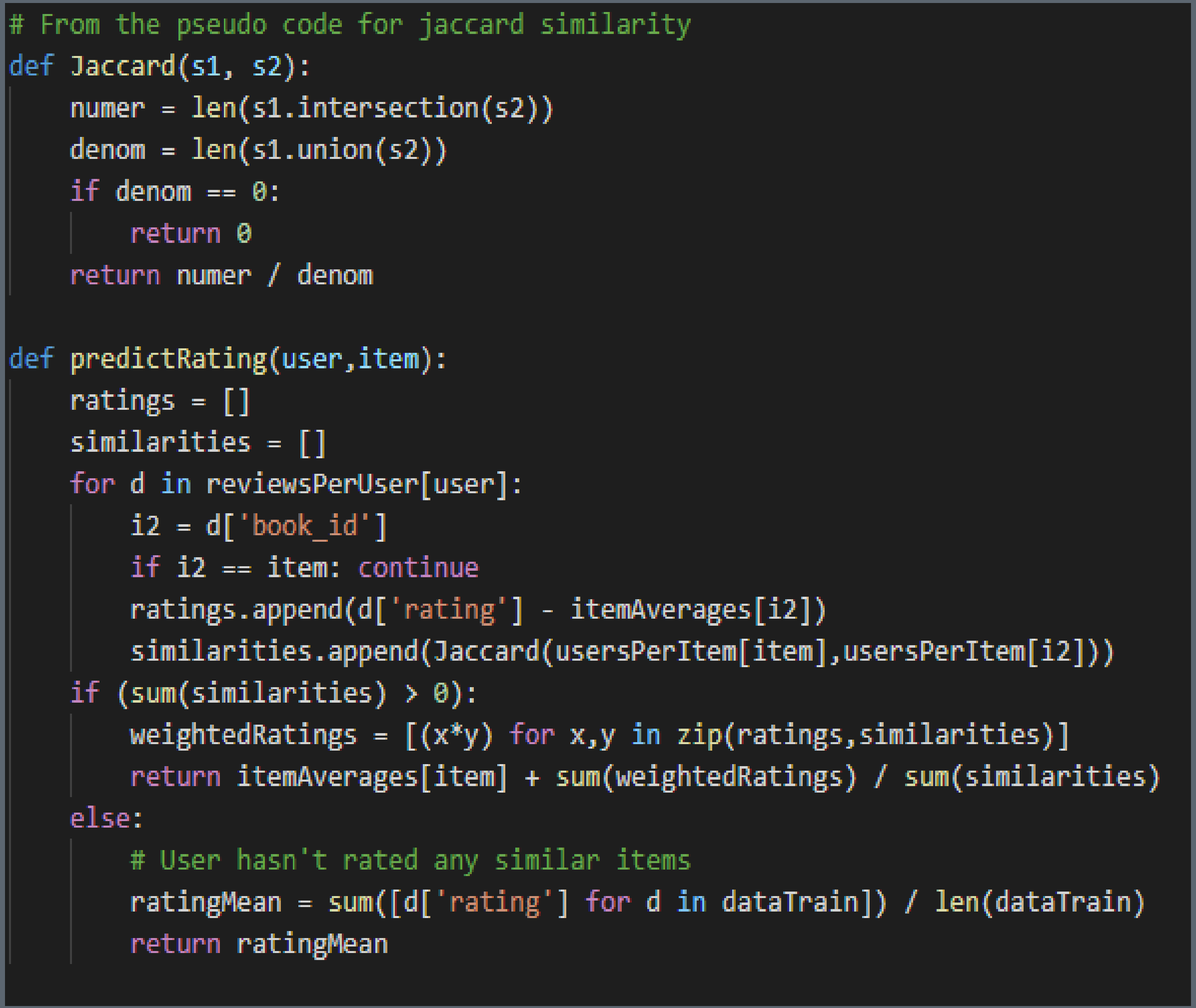
```python
wordCount = defaultdict(int)
punctuation = set(string.punctuation)

for d in dataset:
    r = ''.join([c for c in d['review_text'].lower() if not c in punctuation])
    for w in r.split():
        wordCount[w] += 1

sorted_counts = sorted(wordCount.items(), key=lambda x:x[1])[::-1]
```



```python
def feature(datum):
    feat = [0]*len(words)
    # removing punctuation from review
    r = ''.join([c for c in datum['review_text'].lower() if not c in punctuation])
    # adding word counts to feature
    for w in r.split():
        if w in words:
            feat[wordId[w]] += 1
    feat.append(1)
    return feat
```

Code Walk Through

- Code variables for Jaccard similarity -
  - Users Per Item - Dict for unique item as values and different users as key.
  - Item Per Users - Dict for unique users as values and different item as key.
  - similarities - Jaccard similarities generated from the function <Jaccard>



```python
# From the pseudo code for jaccard similarity
def Jaccard(s1, s2):
    numer = len(s1.intersection(s2))
    denom = len(s1.union(s2))
    if denom == 0:
        return 0
    return numer / denom

def predictRating(user,item):
    ratings = []
    similarities = []
    for d in reviewsPerUser[user]:
        i2 = d['book_id']
        if i2 == item: continue
        ratings.append(d['rating'] - itemAverages[i2])
        similarities.append(Jaccard(usersPerItem[item],usersPerItem[i2]))
    if (sum(similarities) > 0):
        weightedRatings = [(x*y) for x,y in zip(ratings,similarities)]
        return itemAverages[item] + sum(weightedRatings) / sum(similarities)
    else:
        # User hasn't rated any similar items
        ratingMean = sum([d['rating'] for d in dataTrain]) / len(dataTrain)
        return ratingMean
```

Results

Bag of Words Approach

Logistic Regression

- Accuracy - 45.5%

Linear Regression

- Mean squared error - 1.575
- Mean absolute percentage error - 36.61%

Jaccard Similarity

Mean squared error - 1.837

Mean absolute error - 31.01%

Future Refinements

THIS PROJECT IS SOLVING A REAL-WORLD PROBLEM WHERE WE ARE PREDICTING THE FUTURE RATING OF THE USER BASED ON THE PREVIOUS DATA

THIS SAME MACHINE LEARNING MODEL CAN BE USED FOR PREDICTING THE RATINGS OF REVIEWS OF THE HOTELS.

THE MODEL CAN ALSO BE USED TO GIVE THE PRODUCT RECOMMENDATION BASED ON THE RATING OF THE USER ON PREVIOUS PURCHASES AND THE RATING OF THE USER WITH SIMILAR PURCHASE PATTERN

Arizona State University

Thank You