Recysis

Sentiment Analysis

Data Mining and Machine Learning
Project a.a. 2021-2022

Irene Cantini
Elisa De Filomeno

Contents overviews

- Introduction
- Sentiment Analysis
- Streaming Analysis
- Application
- Conclusions

PRIMI PIATTI
Risotto mantecato con zafferano dei Campiani, gamberi rossi, calamari e profumo di liquirizia
Finiture e presentazione
[STRUMENTI E COTTURE]

Introduction

Introduction

- Recysis is a cooking application offering users over 500,000 recipes,
- It allows users to add reviews to each recipe.
- These reviews contain users’ opinion and emotion about recipes.
- Each recipe contains lots of positive or negative comments.

Goal

- Help users in the choice of what recipes to make by looking at the number of positive, neutral and negative comments that has been left on each recipe.
- Automatically classify comments by the polarity of the sentiment as positive, negative or neutral (Sentiment Analysis) .

Datasets

Source: https://www.kaggle.com/irkaal/foodcom-recipes-and-reviews

- First dataset (recipes.csv): is composed by 522.517 recipes and 28 attributes
- Second dataset (reviews.csv): is composed by 1.401.754 comments and 8 attributes

Recipes.csv	Reviews.csv
RecipeId	
Name	
AuthorId	ReviewId
RecipeCategory	RecipeId
RecipeIngredientParts	AuthorId
RecipeInstruction	AuthorName
Calories	Rating
CholesterolContent	Review
FiberContent	DateSubmitted
SugarContent	DateModified
ProteinContent	
…	

Sentiment analysis

STEPS

1. Dataset Preparation and Training Set

2. Text preprocessing, Building Vocabulary and Feature Extraction

3. Classifiers Evaluations

4. Model Selection

Dataset cleaning

- Dataset: reviews.csv
- Removed empty comments
- Removed from comments:

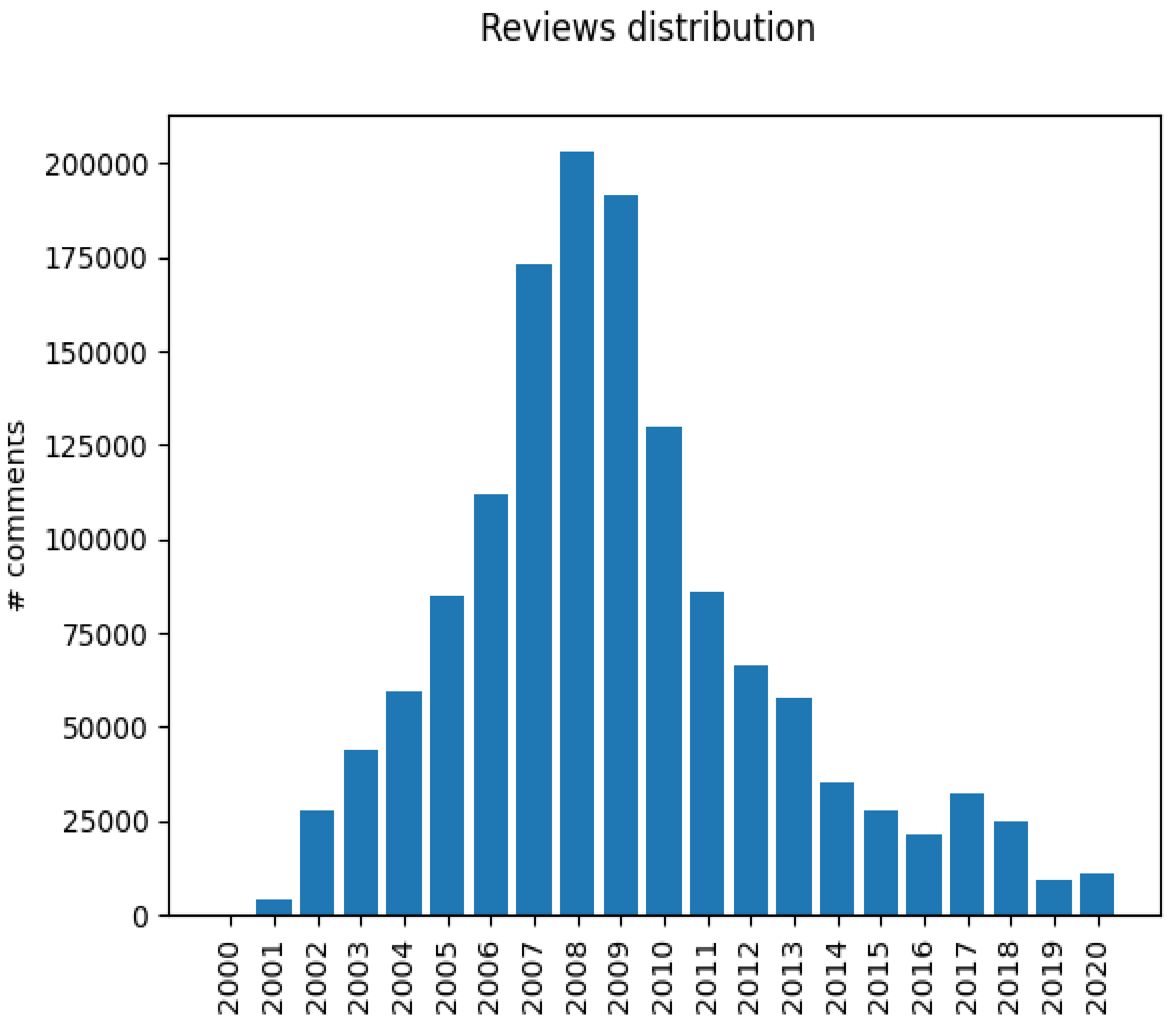
  - characters which represent the end of the line

  - multiple spaces

Data reduction

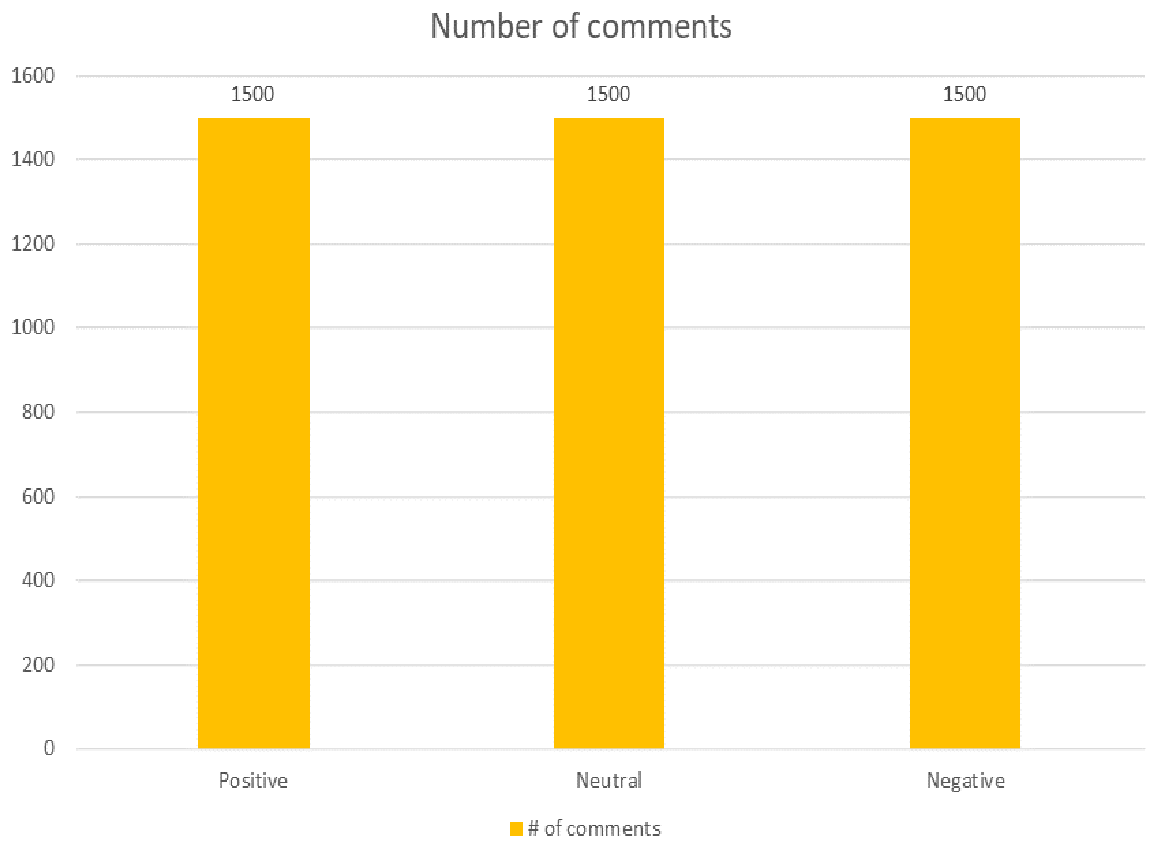
- From1.401.752 comments to 202.979 comments in 2008

- Attribute selected:
  - Review -> containing the text of the comments
  - Rating -> score from 1 to 5, establishing a ground truth



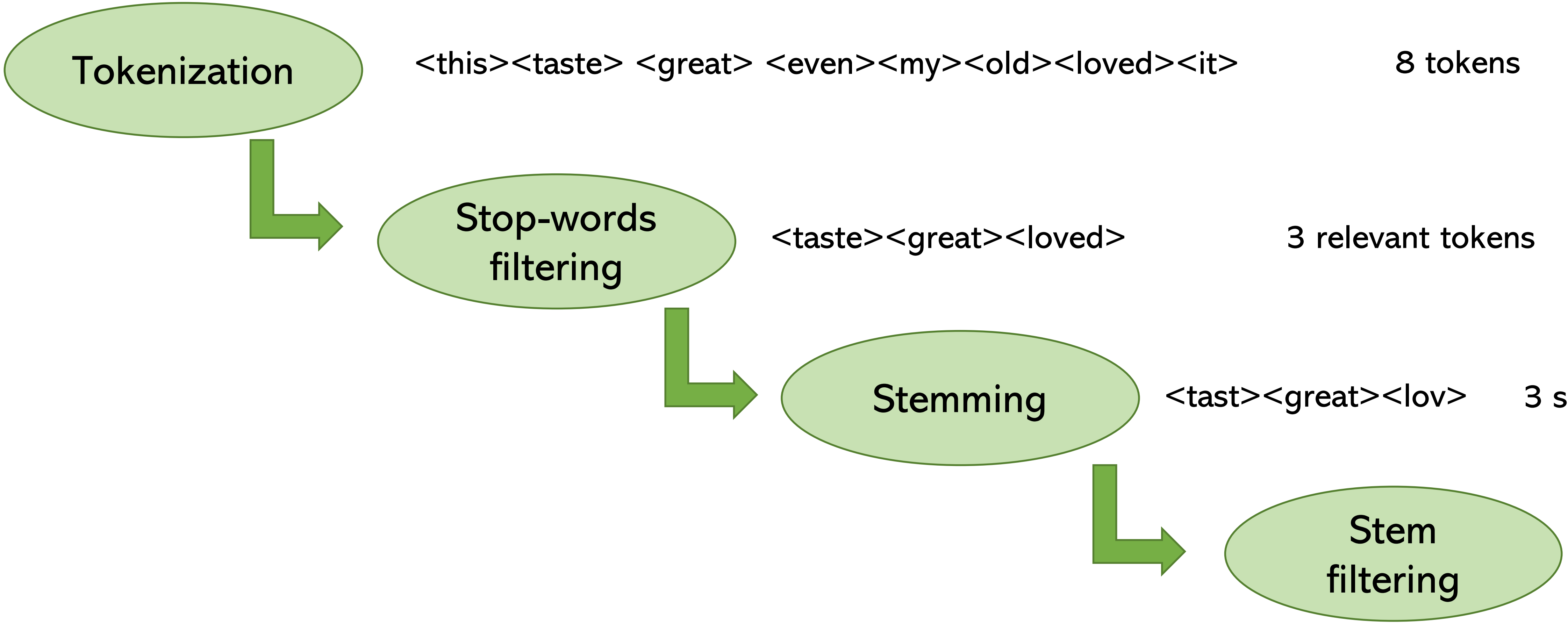
Training Set

- Timeline for training set: 01/01/2008 to 31/12/2008
- 4.500 comments labelled:
  - Rating = 1 → negative comment
  - Rating = 3 → neutral comment
  - Rating = 5 → positive comment
- Balanced training set, 1500 instances for each class



Text Elaboration

“This taste great!!!!!! Even my 3yr. old loved it”



Tokenization

<this><taste> <great> <even><my><old><loved><it>

8 tokens

Stop-words filtering

<taste><great><loved>

3 relevant tokens

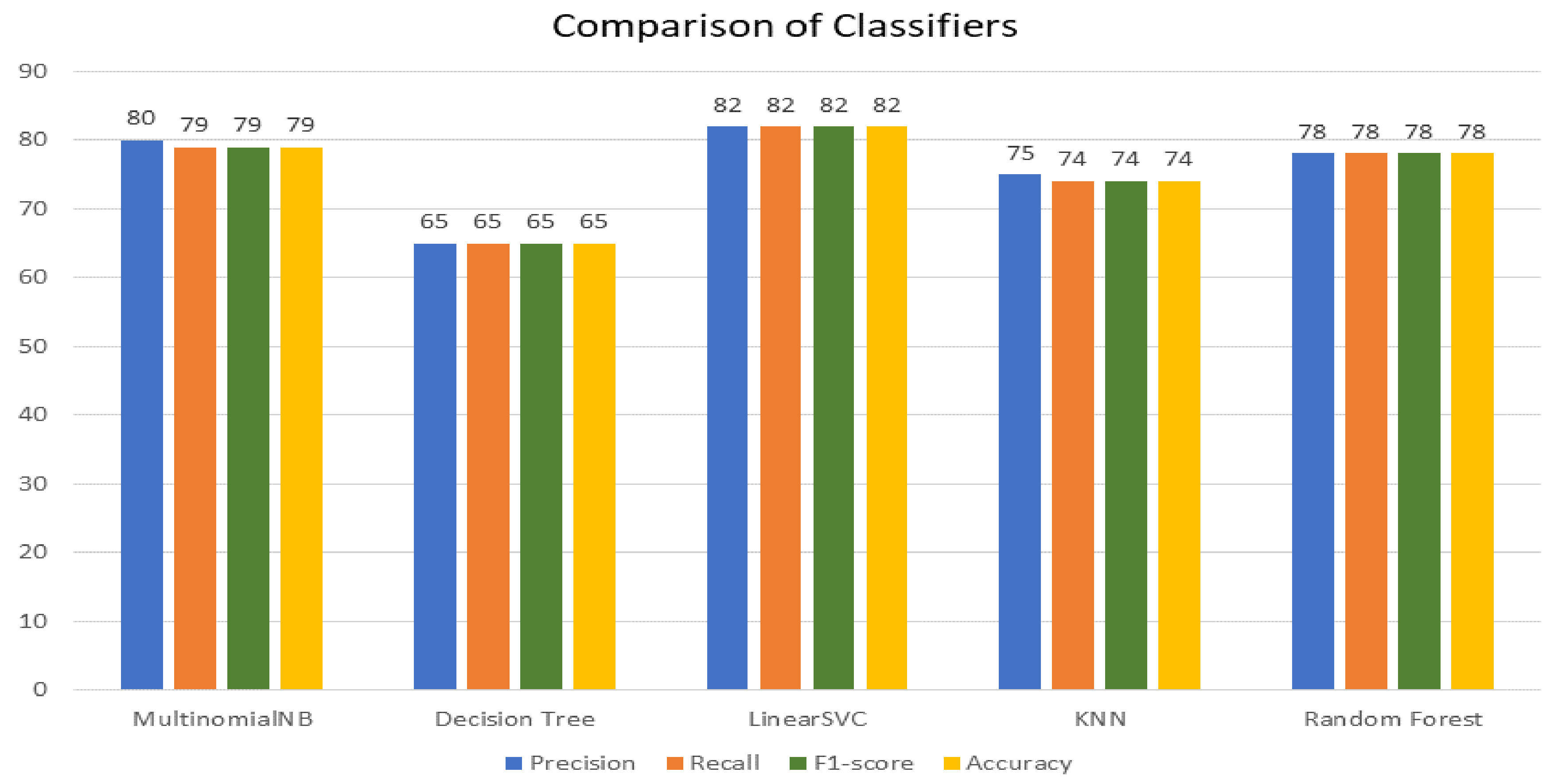
Stemming

<tast><great><lov>

3 stems

Stem filtering

Classification

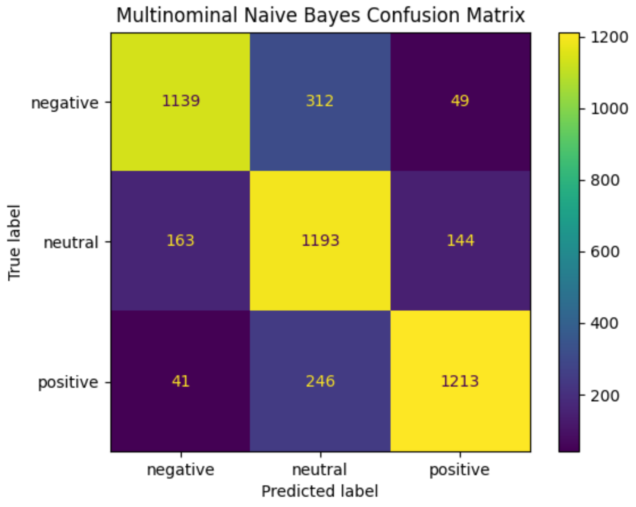


- 5-fold cross validation for each classifier
- Paired T-test between the two best classifiers

Comparison of two best classifiers

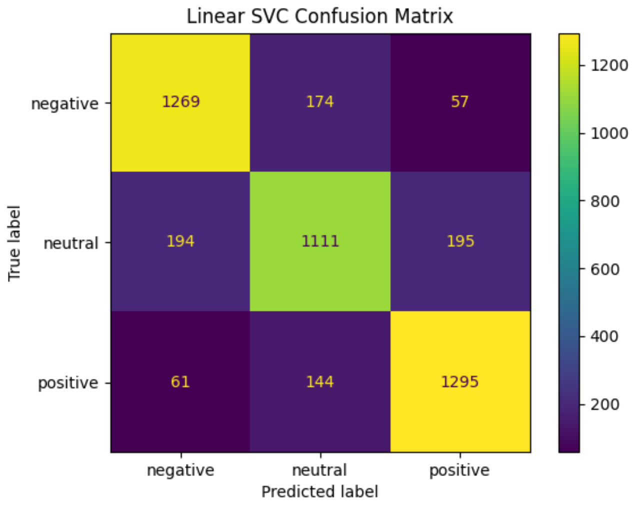
- MultinominalNB()

	Precision	Recall	F1-score	Support
Negative	0.85	0.76	0.80	1500
Neutral	0.68	0.80	0.73	1500
Positive	0.86	0.81	0.83	1500
Average	0.80	0.79	0.79	



- LinearSVC(C=0.1)

	Precision	Recall	F1-score	Support
Negative	0.83	0.85	0.84	1500
Neutral	0.78	0.74	0.76	1500
Positive	0.84	0.86	0.85	1500
Average	0.82	0.82	0.82	



Results

- **t-test**: $\alpha=0.05$, p-value=0.003 → $p < \alpha$
- General high accuracies in classifying positive, negative and neutral comments
- **LinearSVC** is the most performing classifier

	ACCURACY	ERROR RATE
multinominalNB	78.777%	21.222%
linearSVC	81.666 %	18.333%

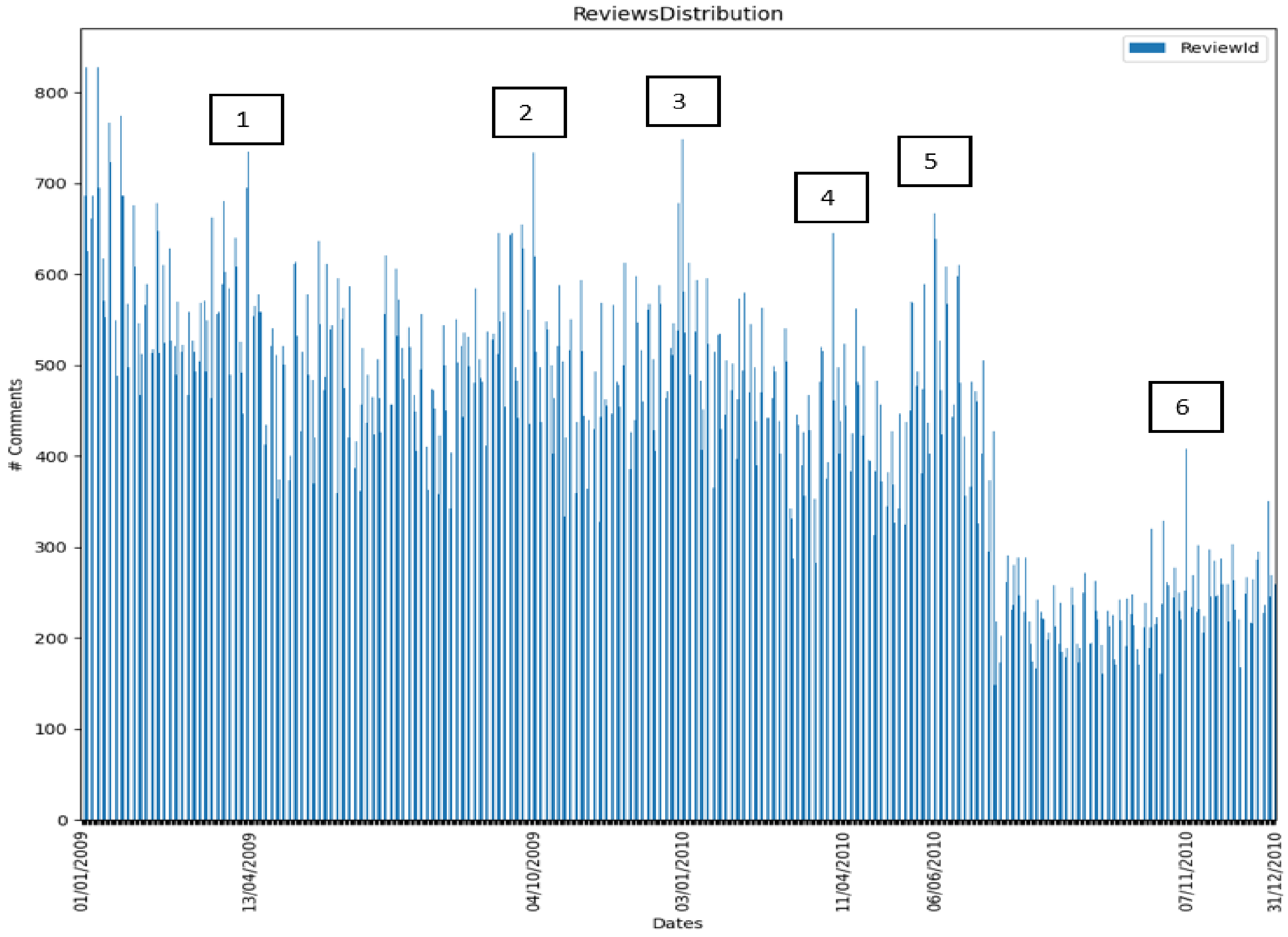
Streaming Analysis

STEPS

1. Select a subsequent time window
2. Select a certan number of events
3. Find comments about these events
4. Build three different models
5. Select most suitable model

Events

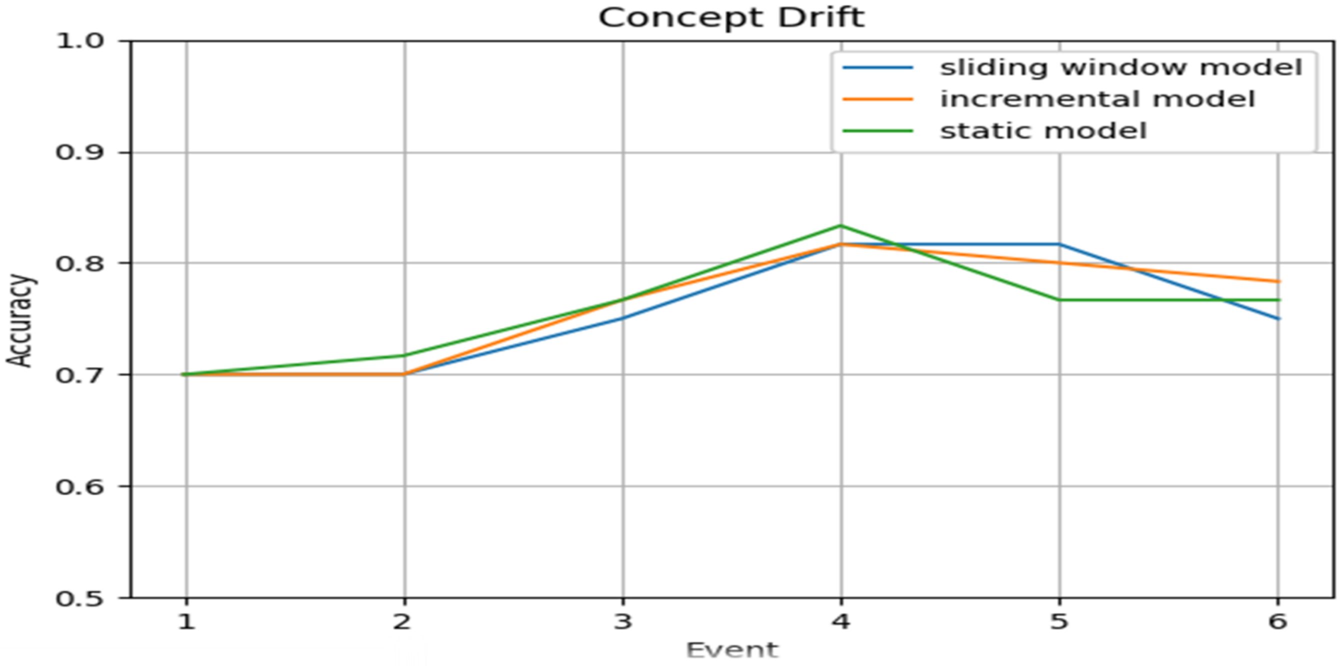
- Subsequent time window (2009-2010)
- Events: Peaks of comments
- Found 6 key events on the timeline

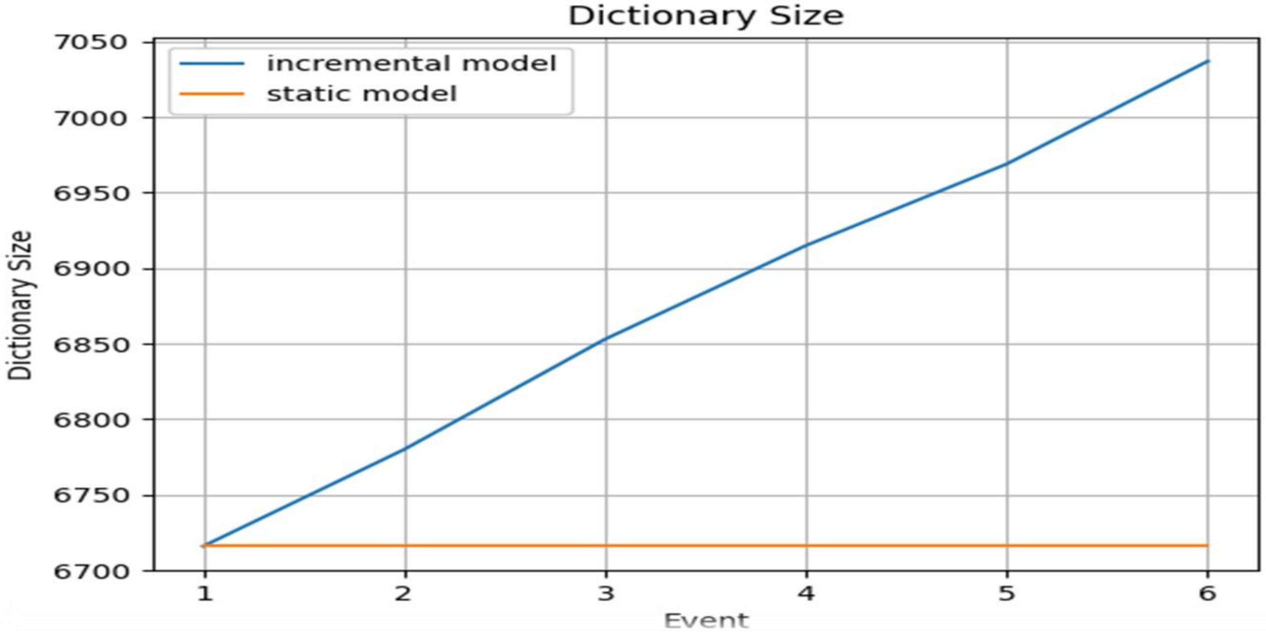


Models

- We implemented a comparative study based on different models
- For each event selected we labelled 60 comments using the Rating attribute as ground truth
- We use those comments as test set for 3 different learning settings:
  - Static model: the initial training set composed by 4500 comments
  - Sliding model: retrained each time with the most recent 4500 comments, removing the oldest 60 and adding the newest 60
  - Incremental model: trained with the initial training set plus all the labelled data of all the previous events before testing on a new event

Results


Concept Drift
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
1
2
3
4
5
6
Event
sliding window model
incremental model
static model

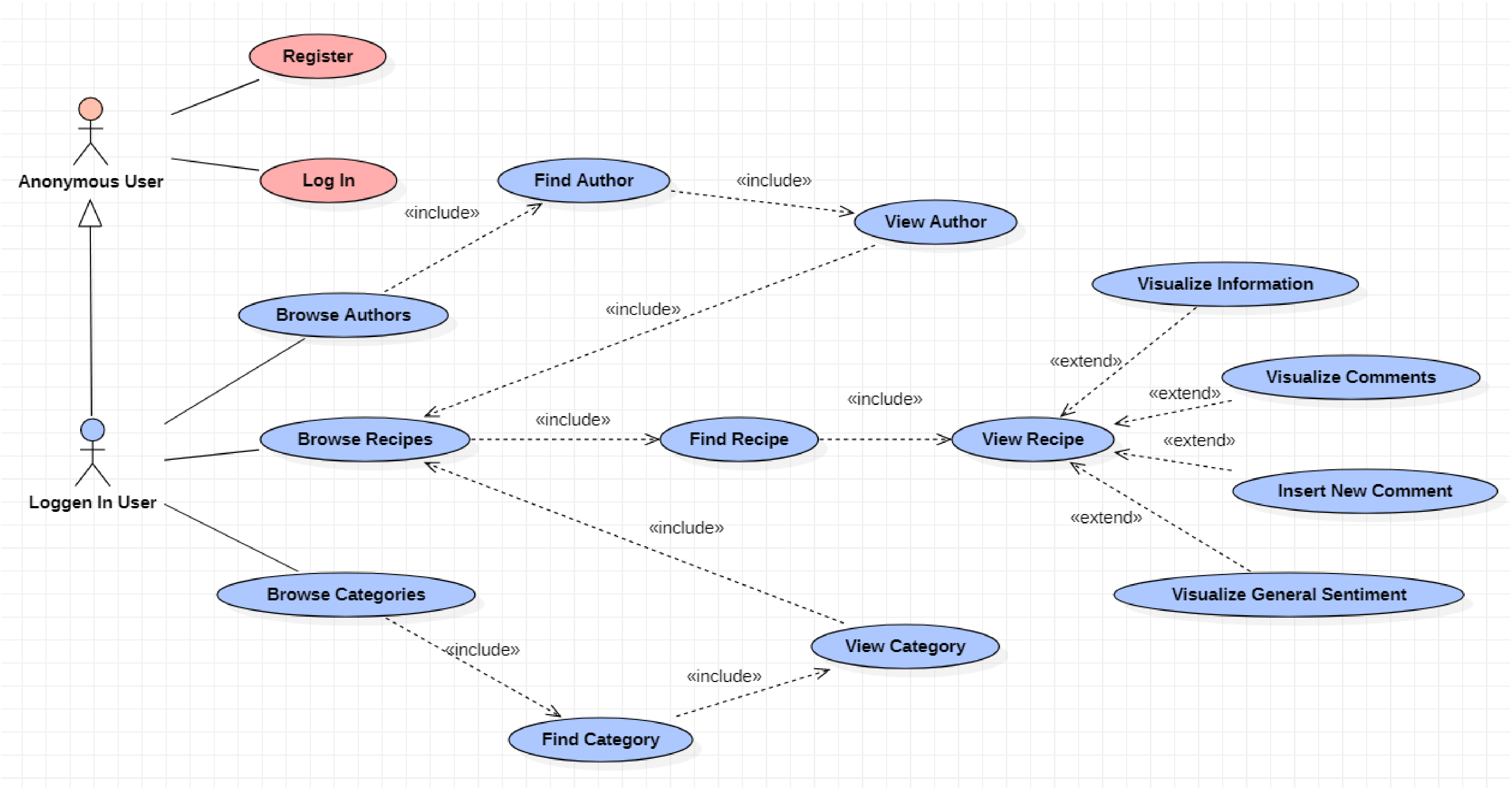

Dictionary Size
Dictionary Size
7050
7000
6950
6900
6850
6800
6750
6700
1
2
3
4
5
6
Event
incremental model
static model

Application

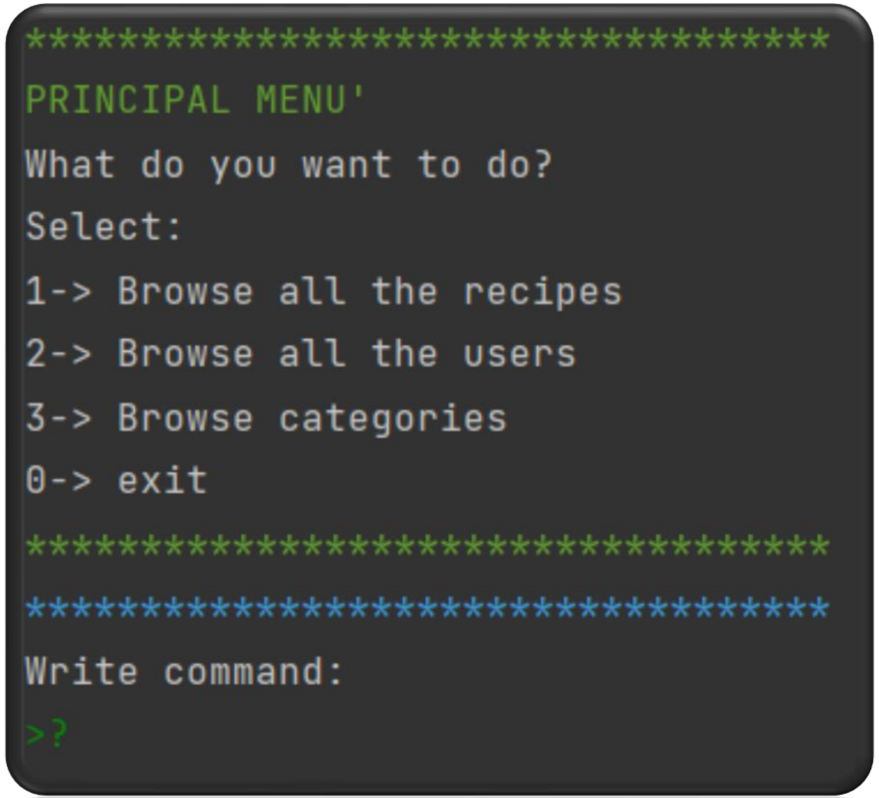
Application

- Command line application developed using Python
- Use the static model
- We embedded in our application all the comments from the original dataset labelled as positive, neutral and negative
- There is also the possibility for the users to add new comments and see immediately the associated sentiment

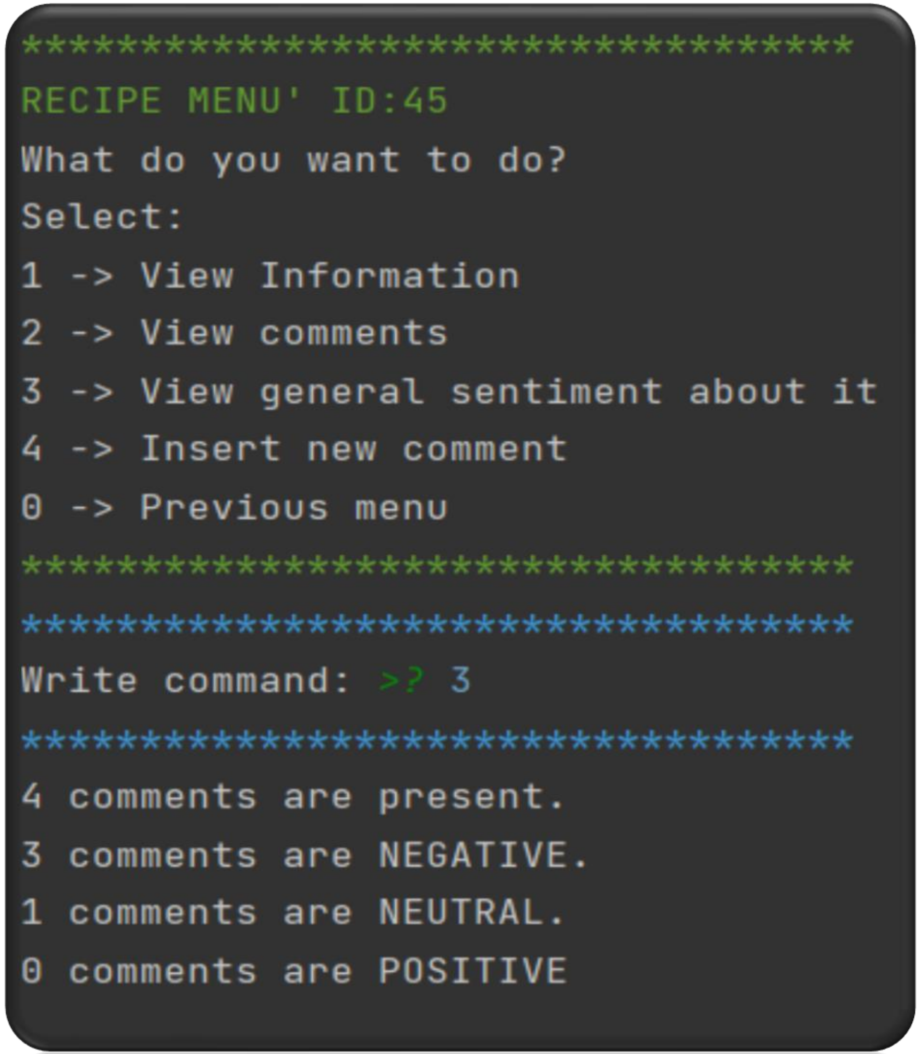
Use case diagram



Application



```
***********************************
PRINCIPAL MENU'
What do you want to do?
Select:
1-> Browse all the recipes
2-> Browse all the users
3-> Browse categories
0-> exit
***********************************
***********************************
Write command:
>?
```



```
***********************************
RECIPE MENU' ID:45
What do you want to do?
Select:
1 -> View Information
2 -> View comments
3 -> View general sentiment about it
4 -> Insert new comment
0 -> Previous menu
***********************************
***********************************
Write command: >? 3
***********************************
4 comments are present.
3 comments are NEGATIVE.
1 comments are NEUTRAL.
0 comments are POSITIVE
```

Conclusions

Conclusions

- The service we provide offers an additional functionalities to explore all the recipes' impressions

- The application gives an immediate snapshot on the goodness of a recipe useful for both final users and the recipe's owner

THANKS FOR YOUR ATTENTION!

Data Mining and Machine Learning
Project a.a. 2021-2022

Irene Cantini
Elisa De Filomeno