Games classifier

Team name

Members: Julia Cygan, Borys Adamiak, Patryk Flama
Supervisor: Marek Adamczyk

UWr

January 27, 2025

Goal and motivation

Use case example:
Imagine that you run a online game store where users can add their games to your library. Instead of manually checking if user tagged correctly the game, you can use our model to do that job for you.

Goal:
We want to be able to automatically assign tags (or genres) to games, based on their (text) description.

Additionally, in aspect of ML project, we want to make a small comparison of different models and methods for solving such multilabel classification problem.

Info about the dataset

Steam has its own official API, from which we downloaded games, their descriptions, tags and genres. That resulted in a bit over *200'000* games.

To clean the data we:

- Converted descriptions to alphanumeric lowercase
- Removed html tags
- Removed empty descriptions or tags
- (optional) Removed tags/genres that occured at most n times

After that we ended up with a dataset of size around *50'000* games and *400* unique tags or *100* unique genres.

Data preprocessing

To represent the output we decided to use multi label binary vector.

Data preprocessing

To represent the output we decided to use multi label binary vector.

For input preprocessing we tried:

- Bag of Words
- TF-IDF
- Hashing vectorizer

Data preprocessing

To represent the output we decided to use multi label binary vector.

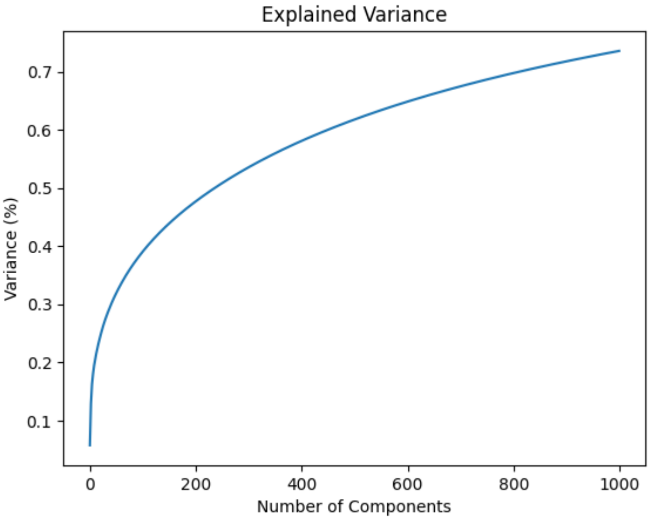
For input preprocessing we tried:

- Bag of Words
- TF-IDF
- Hashing vectorizer

We decided to check if there are some patterns in the data that we can use to improve our model.

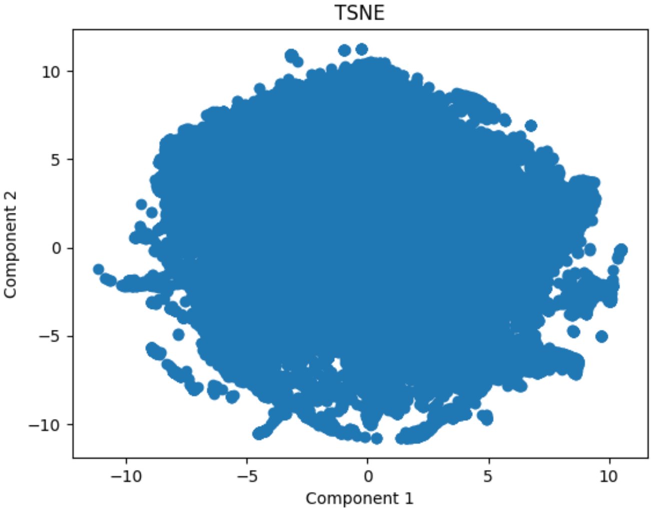
Data preprocessing

Figure: PCA analysis on Bag of Words



Data preprocessing

Figure: t-SNE 300 iterations + PCA to 50 on Bag of Words



Models

- KNN ✓
- Logistic Regression ✓
- Decision Trees + Random Forest ✓
- Naive Bayes ✓
- Simple perceptron-based neural network ✓
- Support Vector Machine ✓

Evaluation

After analyzing the problem we came to conclusion that evaluaion methods are very interesing and important part of this project.

Evaluation

After analyzing the problem we came to conclusion that evaluaion methods are very interesing and important part of this project.

- We dont want to falsely assign a tag to a game that should not have it
- Its more important to assing high percentage of tags to games, than to assign as many as possible
  - Game should have 10 tags, but we only assign 8 (not bad)
  - Game should have 1 tag, but we do not assign any (this is worse)

Evaluation

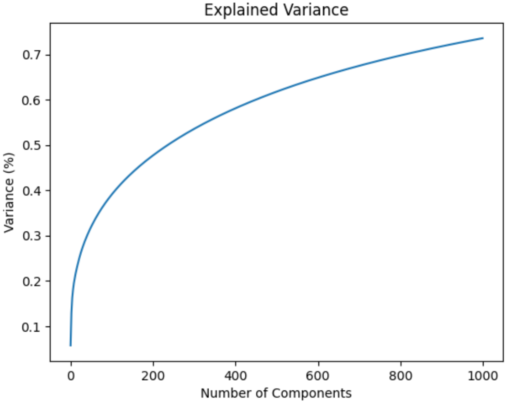
- Recall $TP/(TP+FN)$ - we prefer to have more FN than to have an TP ✓
- F1-score $(2 * \mathit{precision} * \mathit{recall}) / (\mathit{precision} + \mathit{recall})$ - nice name, but also it combines precision with recall thus both TP and FN are equally expensive ✓
- Hammming loss
- Intersection over union score
- Exact match

Input data

First we tried some unsupervised methods to check if we can find some patterns in the data

Input data

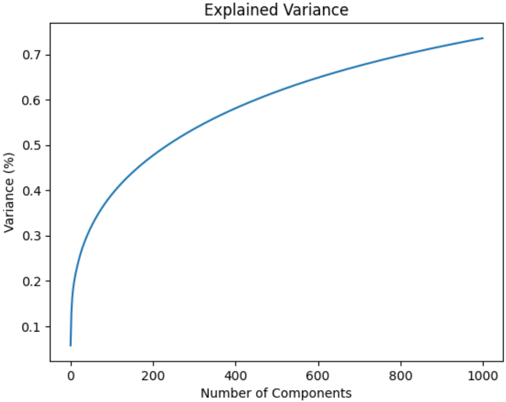
PCA on Bag of Words representation of the data (to 10'000 words)



Nice, out of 10'000 dimensions we can create 100 that 'explain' about half of the data.

Input data

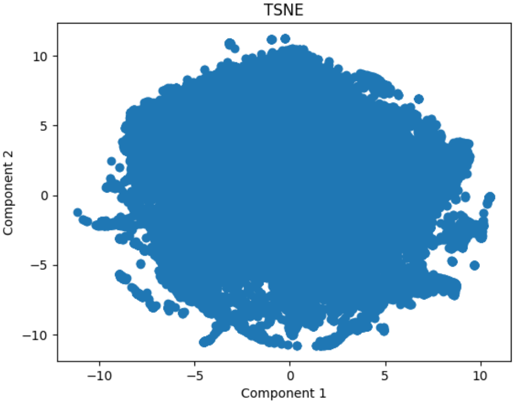
PCA on Bag of Words representation of the data (to 10'000 words)



Nice, out of 10'000 dimensions we can create 100 that 'explain' about half of the data.

Input data

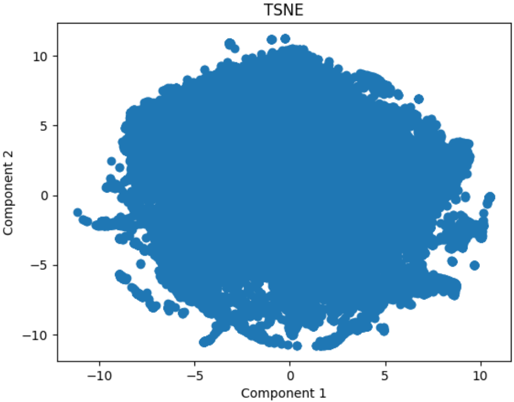
t-SNE representation of same data



And it does not look that helpful :<

Input data

t-SNE representation of same data



And it does not look that helpful :<
But what about combining PCA with t-SNE?

Input data

But what about combining PCA with t-SNE?