Movie Genre Classification

High-dimensional data analysis and Machine Learning

First part: Anne Ruiz-Gazen

Druilhe - Larose - Saue - Steunou - 02/2024

« A superhero battles a chaotic villain to save Gotham City »

The Dark Knight (2008)

« Toys come to life when humans aren't around, embarking on daring adventures. »

Toy Story (1995)

Our goal ?

Create a Machine Learning model that will be able to predict the genre of a movie based on a text description.

Data

Initial Data

IMDb (Internet Movie Data Base)

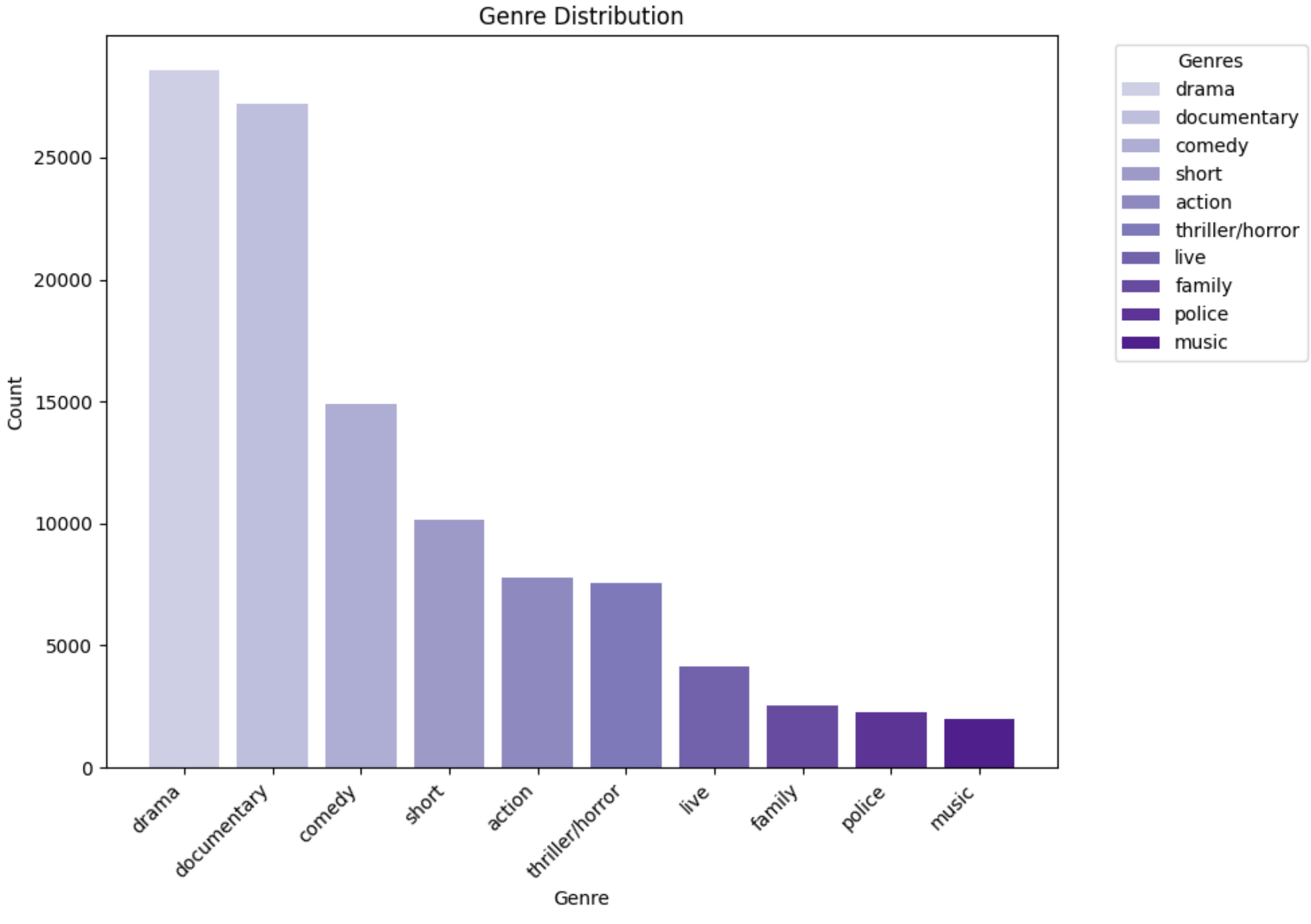
—> 107k Observations

—> Variables:

- Title
- Genre
- Description

Distribution of Movie Genre

Ten different genres



Data pre-processing

Data pre-processing goal

Transform textual data into tabular data

→ Tokenization

→ Vectorization (word2vec)

Tokenization

- Tokenization breaks down text into smaller units called tokens.
- Tokens can be words, phrases, or symbols.
- Extract important words by filtering out common stop words *(such as « the », « and », « ! » and « is »)*.

Tokenization

—> Tokenization optimizes analysis by structuring data and isolating meaningful terms.

Ex: « *The life of penguins !* » —> [‘life’, ‘penguins’]

Vectorization

- Vectorization converts text into numerical vectors, representing words or phrases in a high-dimensional space.
- Word2Vec is Neural Network model that maps words to continuous vectors based on their contextual usage in a text.
- This process captures semantic relationships between words, enhancing analysis and modeling capabilities.

Vectorization

<u>Ex:</u> [‘life’, ‘penguins’]

—> « life » : [0.23, -0.45, 0.67, ...] (100-d vector)

—> « penguins": [-0.12, 0.56, -0.78, ...] (100-d vector)

Final Dataset

	title	genre	description	description_t	embedding	embedding_0	embedding_1	embedding_2	embedding_3	embedding_4	...
0	Oscar et la dame rose (2009)	drama	Listening in to a conversation between his do...	['listening', 'conversation', 'doctor', 'paren...	[0.78826123 0.32318643 -0.04607033 0.520576...	0.788261	0.323186	-0.046070	0.520577	-0.754961	...
1	Cupid (1997)	thriller/horror	A brother and sister with a past incestuous r...	['brother', 'sister', 'past', 'incestuous', 'r...	[1.1432536 0.79619163 -0.18103665 -0.429090...	1.143254	0.796192	-0.181037	-0.429091	-0.572901	...
2	The Secret Sin (1915)	drama	To help their unemployed father make ends mee...	['help', 'unemployed', 'father', 'ends', 'meet...	[3.71534109e-01 3.26947391e-01 -4.91340280e-...	0.371534	0.326947	-0.491340	-0.152857	-0.608980	...
3	The Unrecovered (2007)	drama	The film's title refers not only to the un-re...	['film', 'title', 'refers', 'un', 'recovered',...	[-0.17556079 0.30246285 0.26503348 1.010264...	-0.175561	0.302463	0.265033	1.010264	0.289771	...
4	Quality Control (2011)	documentary	Quality Control consists of a series of 16mm ...	['quality', 'control', 'consists', 'series', '...	[-0.09106552 -0.03001219 0.37830314 0.407525...	-0.091066	-0.030012	0.378303	0.407525	0.211869	...
...	...	...	...	...	...	...	...	...	...	...	...

—> We end up with a dataset of size *107,234 x 105*

Principal Component Analysis

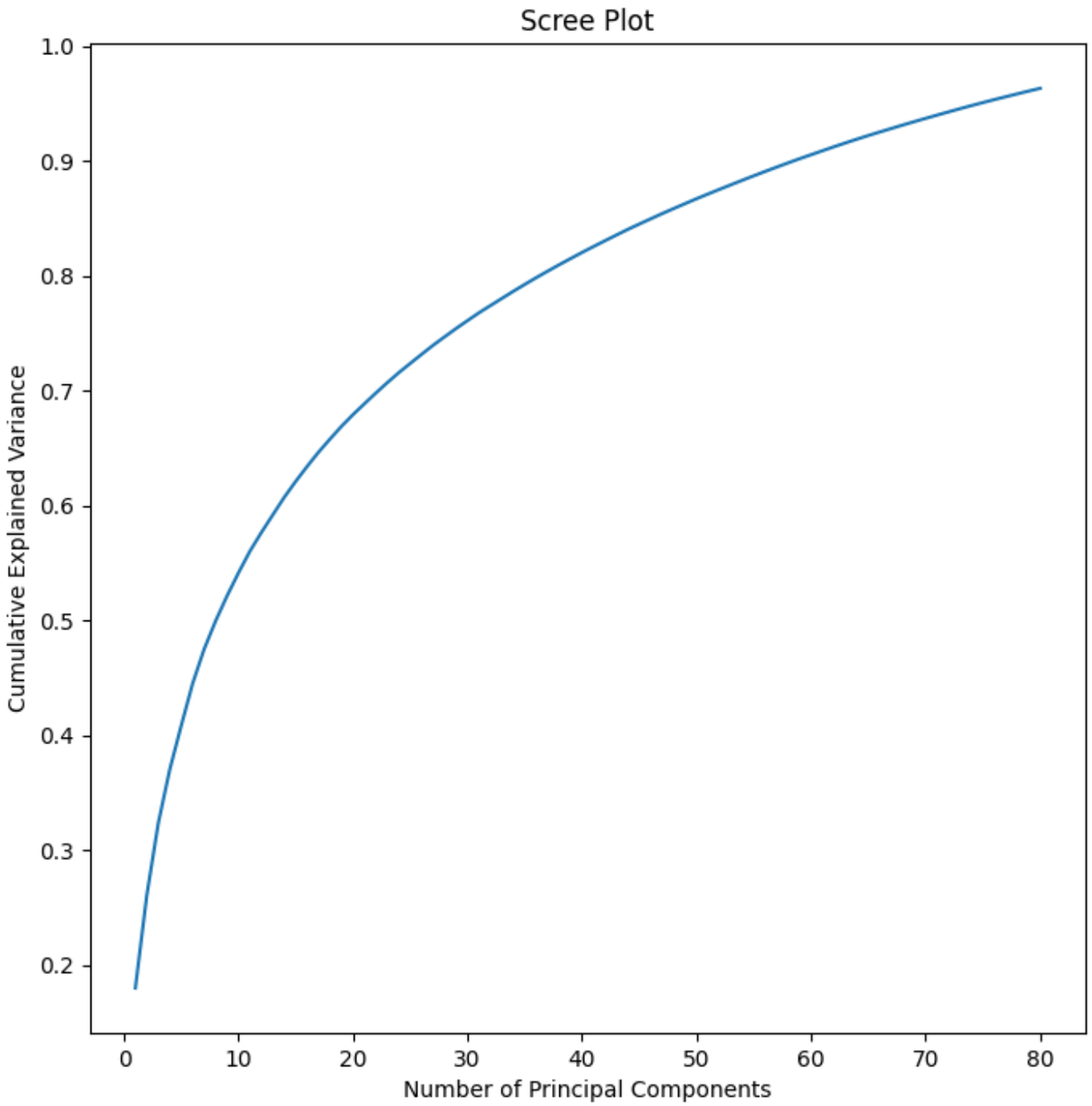
Motivation

—> Mitigates Sparsity

—> Improved Generalization

—> Computational Efficiency

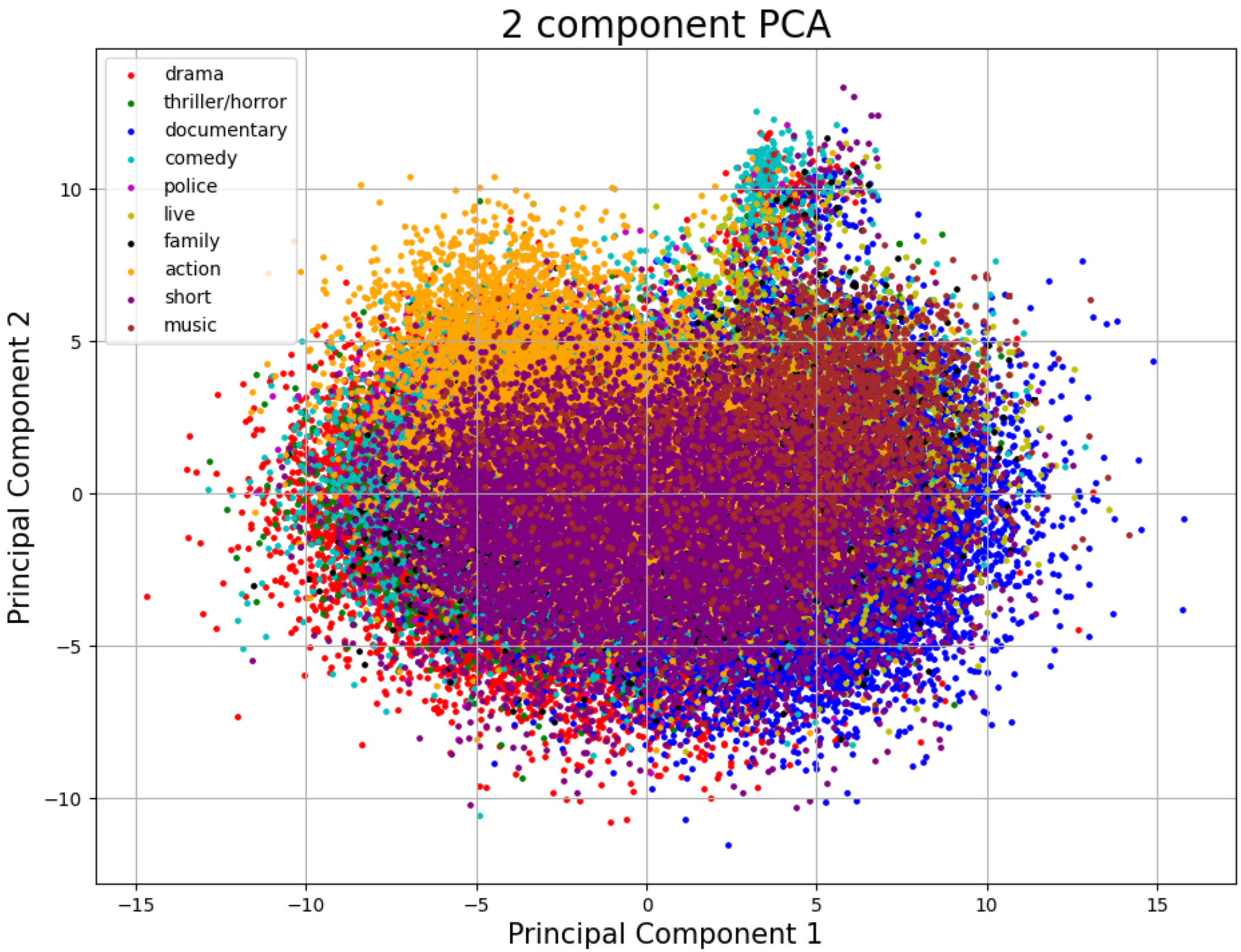
PCA Scree Plot



—> 37 components

—> 80% of the variance

PCA in two dimensions



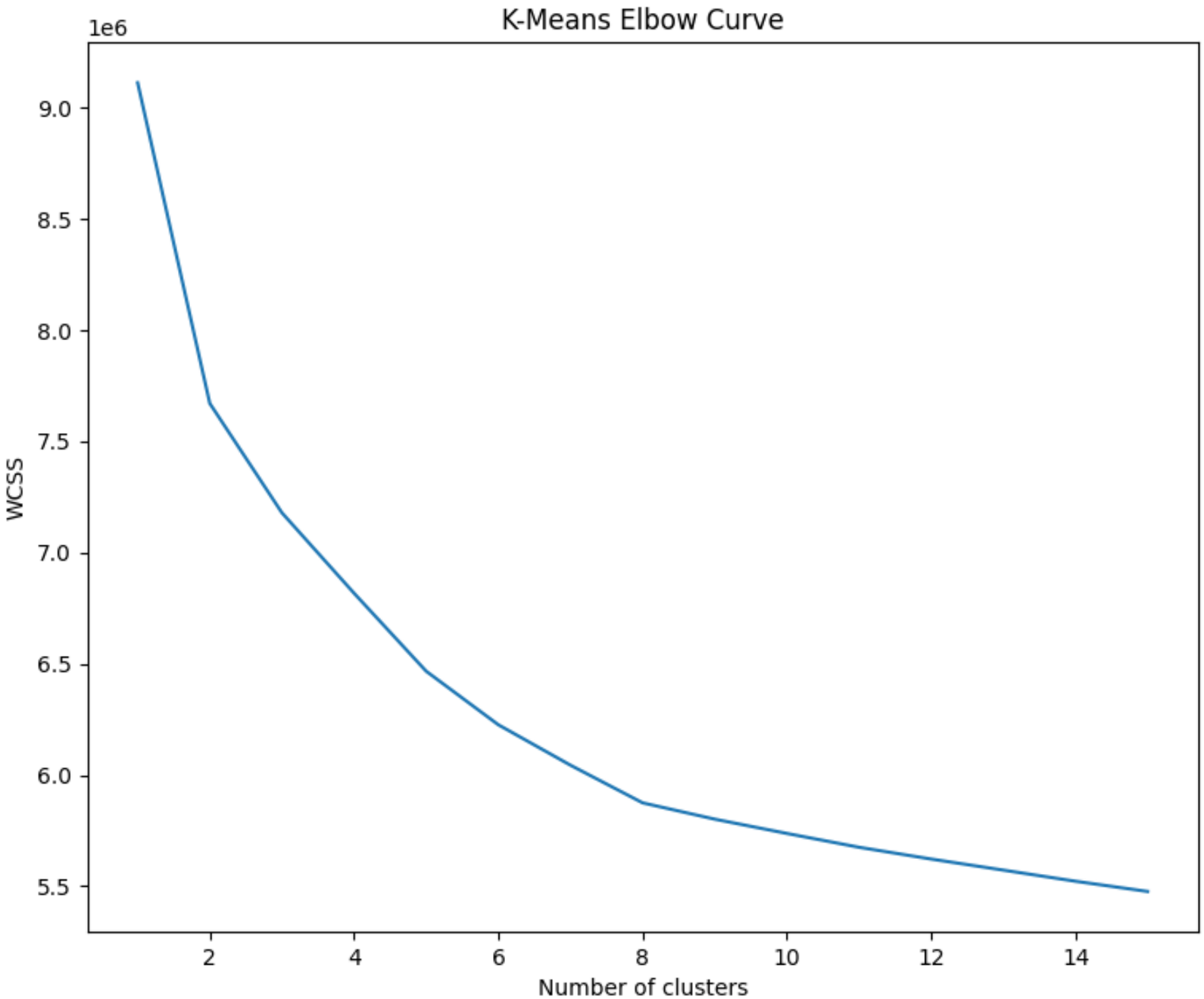
Clustering: K-means

Motivation:

—> Find through the algorithm similarities between genres and/or underlying semantic links between groups of movie descriptions

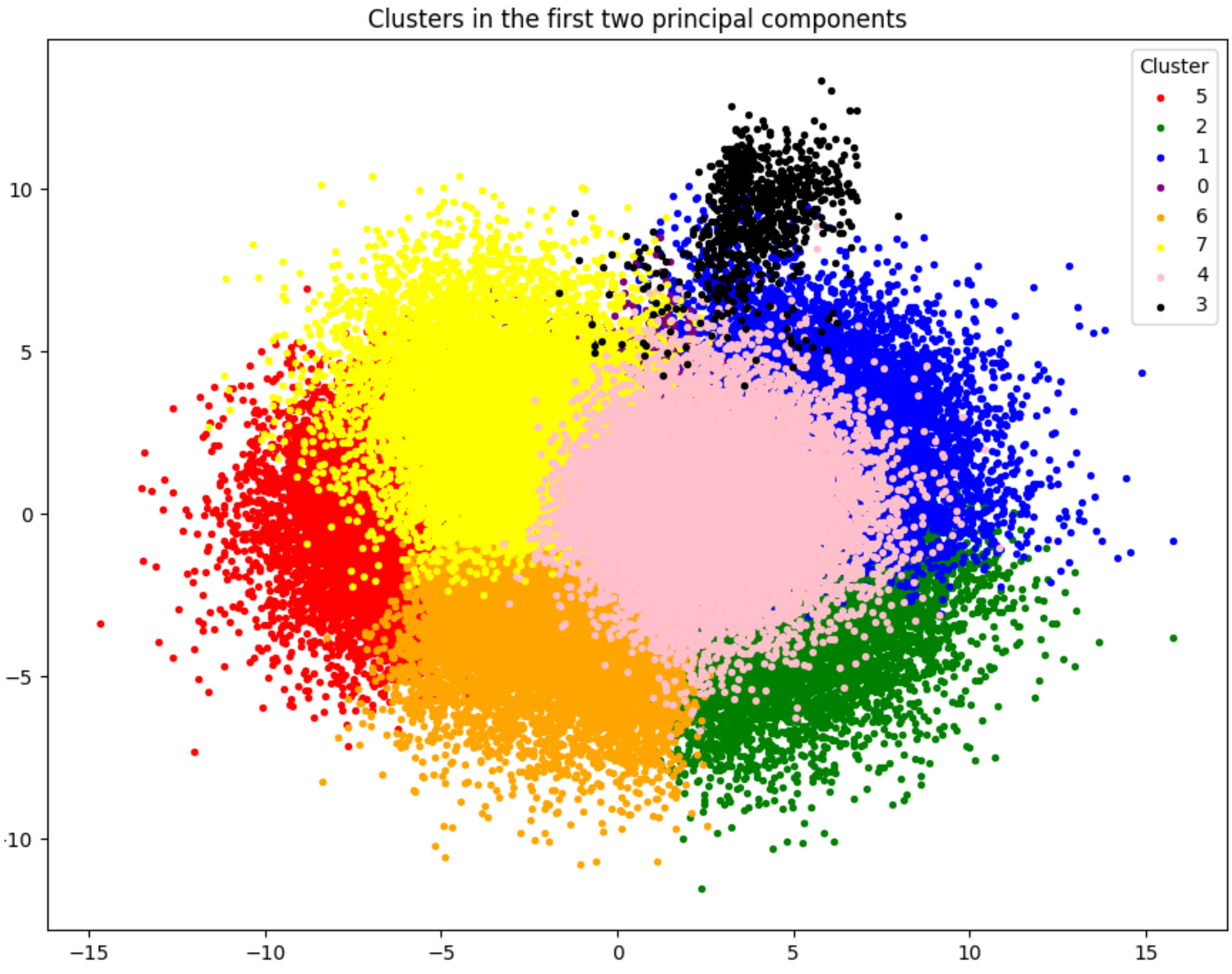
—> Basically, the interpretation of clusters will give us such ties

K-means Elbow Curve



—> Inflexion point at 8 clusters

Clusters Visualization



Interpretation

—> one cluster stands out as being lightly populated

Cluster Number	Number of Movies
0	795
2	12806
5	12975
4	14172
7	14570
6	16186
1	16620
3	19110

Interpretation

—> It is interesting to look at and interpret the top value for each row and column

genre	action	comedy	documentary	drama	family	live	music	police	short	thriller/horror
cluster										
0	2.65	36.28	12.90	29.58	2.40	2.28	0.88	1.01	10.11	1.90
1	4.19	7.76	12.32	45.14	2.26	0.84	0.34	2.62	13.09	11.43
2	2.26	9.85	42.99	5.18	3.08	14.30	10.66	0.61	9.47	1.60
3	2.68	21.81	1.21	55.20	2.19	0.30	0.63	1.67	6.98	7.34
4	1.75	3.49	59.66	10.75	1.43	4.50	0.83	0.83	15.79	0.98
5	32.56	11.89	2.00	21.05	1.90	0.39	0.26	6.64	3.67	19.64
6	4.23	33.83	9.80	20.36	4.67	7.19	1.67	1.62	10.60	6.02
7	7.71	2.52	61.98	14.18	1.03	1.80	0.30	1.45	6.26	2.77

Classification and Regression Tree

Motivation

—> Run a classification algorithm that is flexible when it comes to data type and does not assume linear separation between classes

Results

Accuracy on test set:

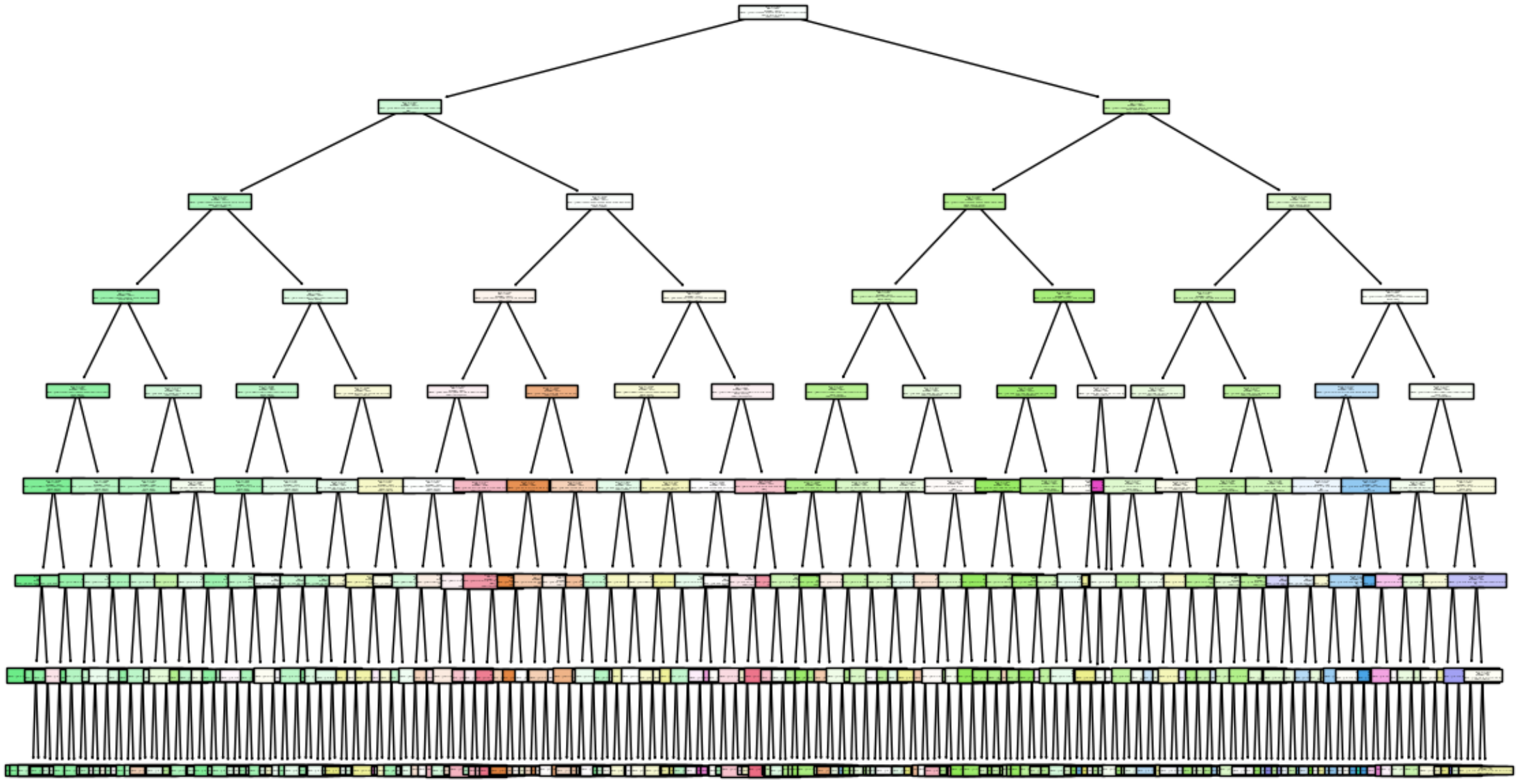
—> Base data: *0.31*

—> After pca: *0.43*

—> After pca and clustering: *0.43*

—> After pca, clustering and pruning: *0.53*

CART: resulting tree



Linear Discriminant Analysis

Motivation:

—> Powerful for textual data analysis and high-dimensional data such as embeddings

—> Finds a projection that maximizes the separation between different classes

—> Utilizes class labels to guide the dimensionality reduction process. Enhances the performance of subsequent classification models.

Results

Class	Precision	Recall	F1-Score	Support
action	0.57	0.54	0.55	1521
comedy	0.55	0.47	0.51	2954
documentary	0.73	0.77	0.75	5574
drama	0.60	0.71	0.65	5714
family	0.39	0.25	0.30	509
live	0.43	0.52	0.47	794
music	0.36	0.59	0.45	397
police	0.26	0.12	0.16	465
short	0.49	0.30	0.37	2008
thriller/horror	0.56	0.57	0.57	1511
accuracy			0.60	21447
macro avg	0.49	0.48	0.48	21447
weighted avg	0.59	0.60	0.59	21447

Table 6: Classification Report on Linear Discriminant Analysis

Results

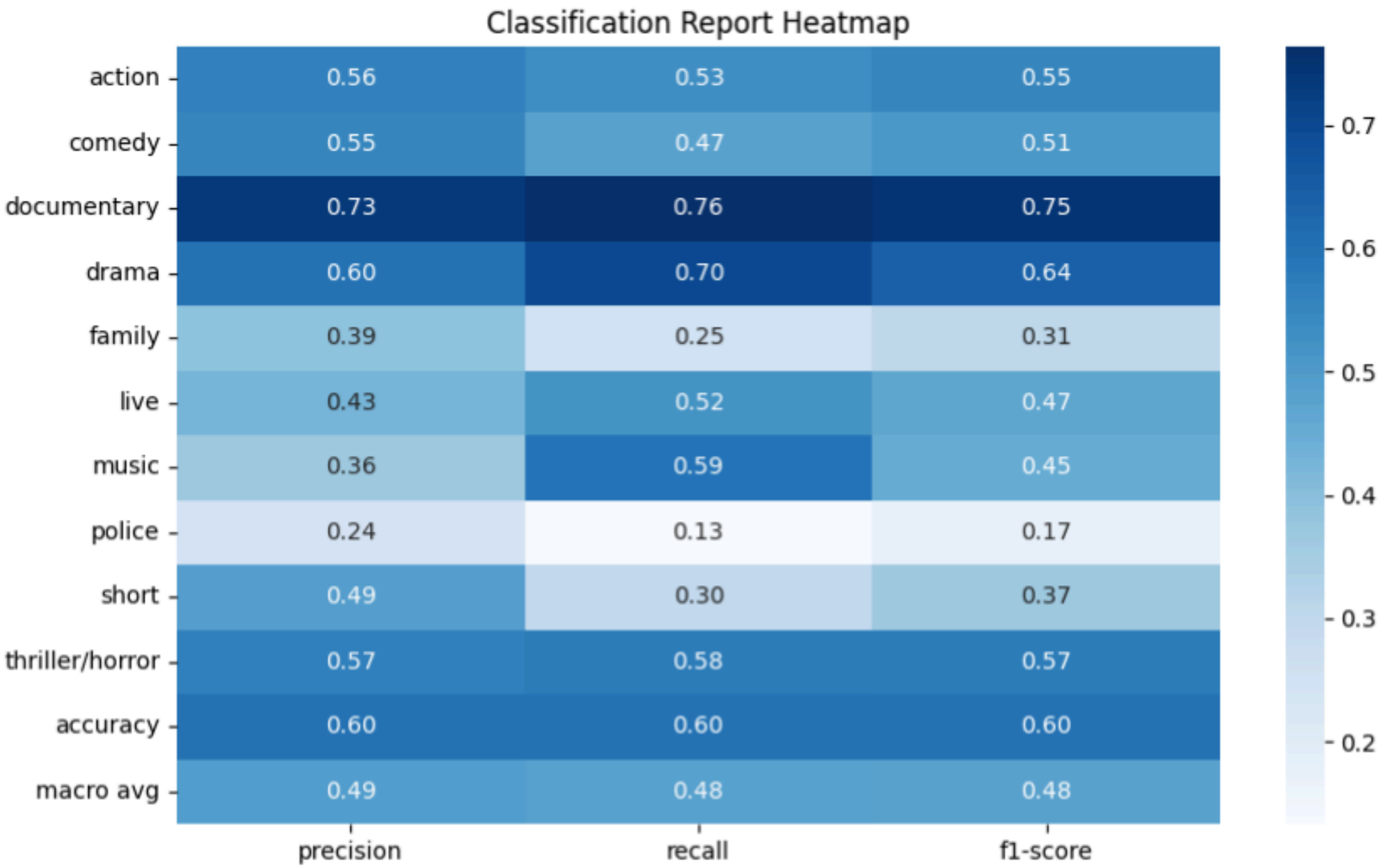
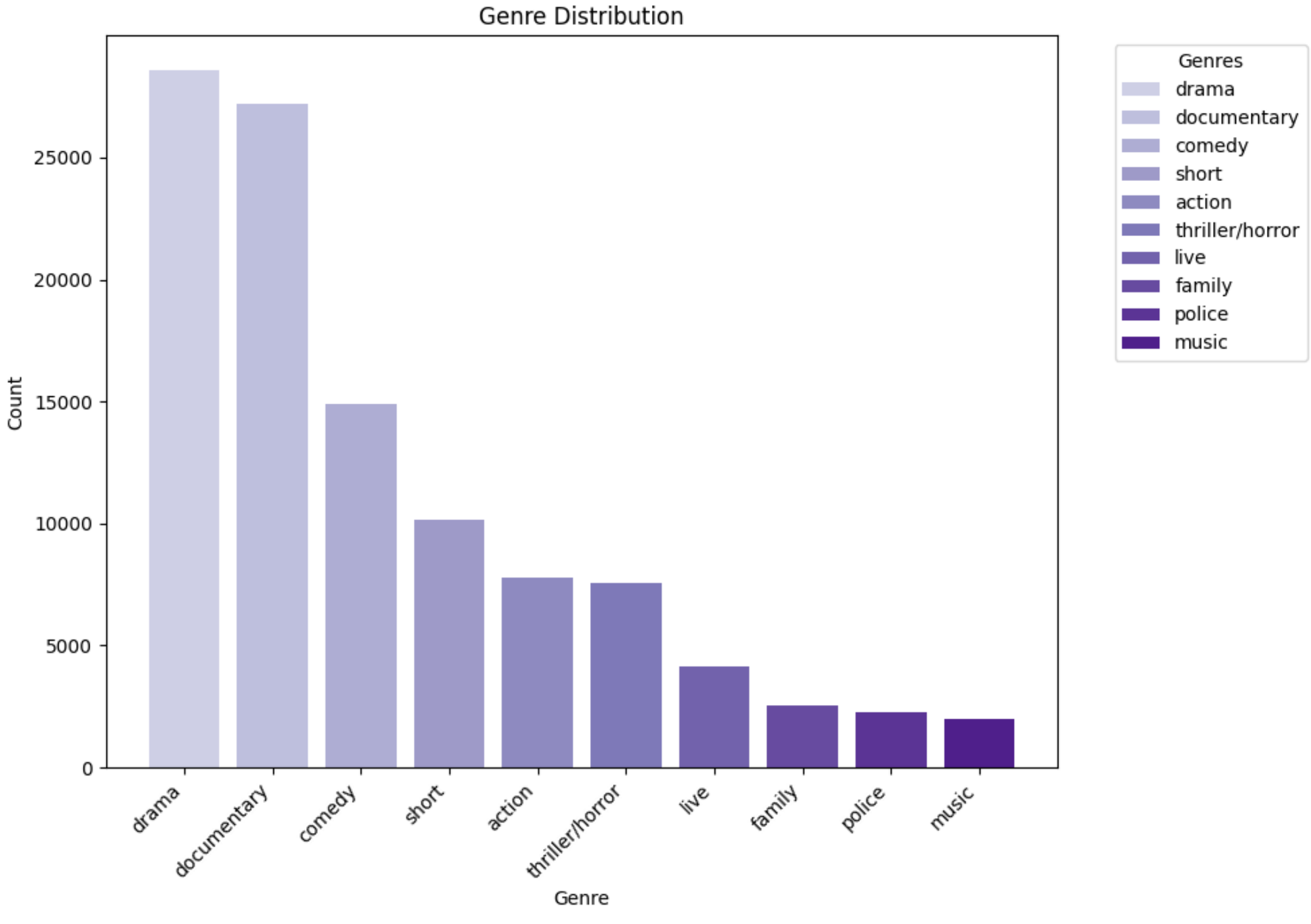


Figure 5: Heatmap for Discriminant Analysis classification result

Distribution of Movie Genre

Ten different genres



Class maps

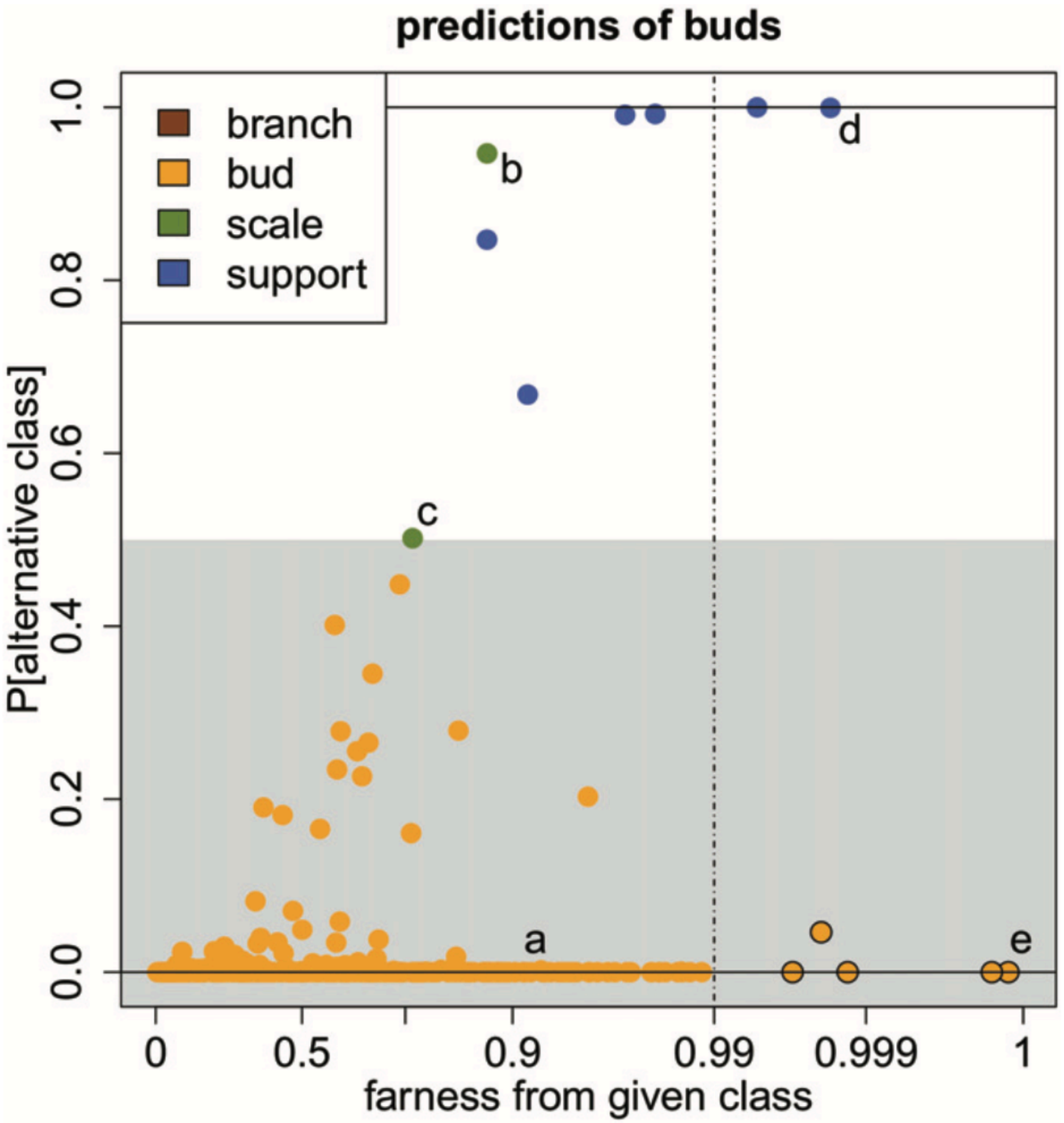
Motivation

—> Shows:

- probability that an object belongs to an alternative class
- how far it is from the given class
- if objects lie far from all classes

—> We use it for discriminant analysis

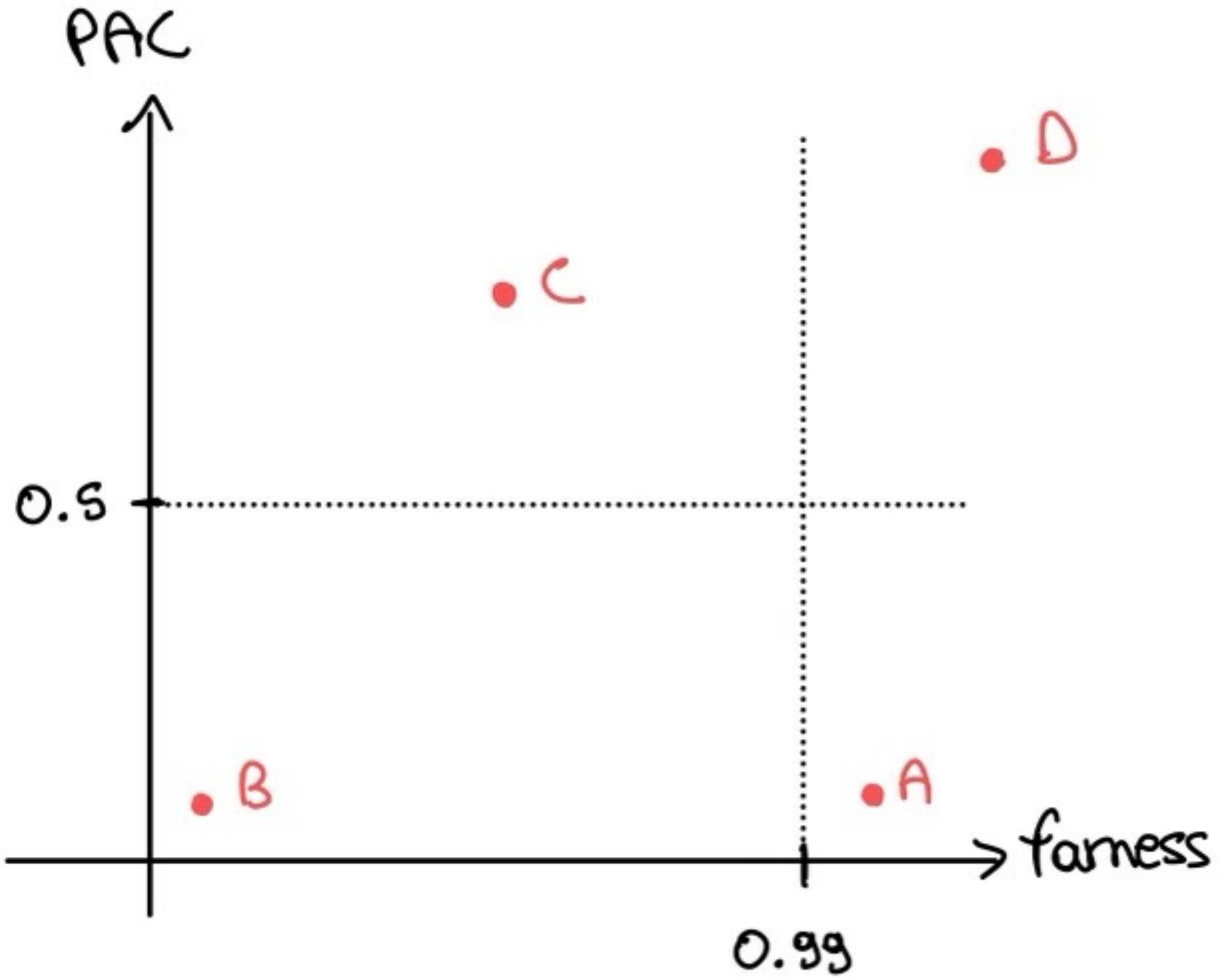
Example

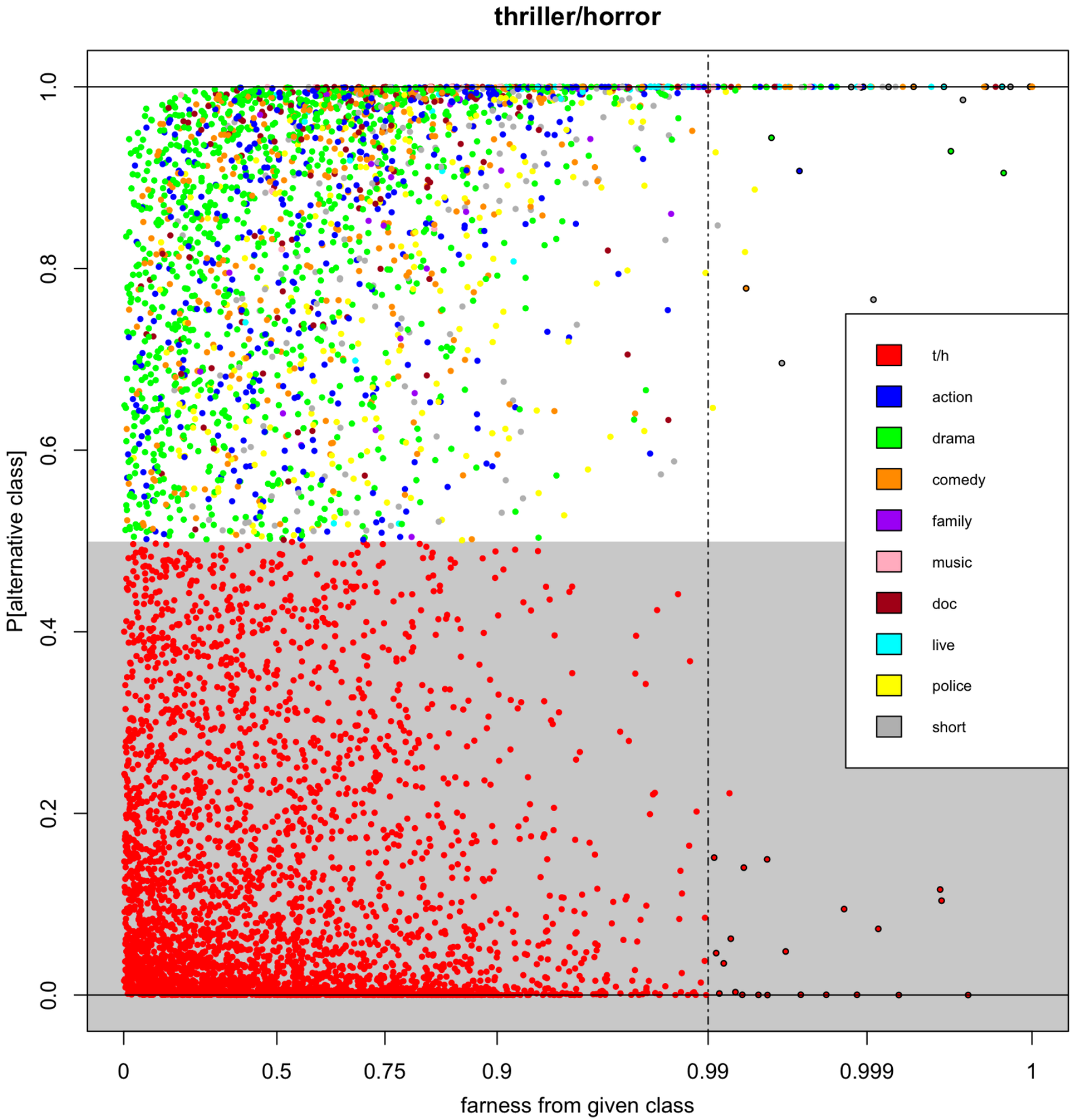

predictions of buds
branch
bud
scale
support
P[alternative class]
1.0
0.8
0.6
0.4
0.2
0.0
0
0.5
0.9
0.99
0.999
1
farness from given class
a
b
c
d
e

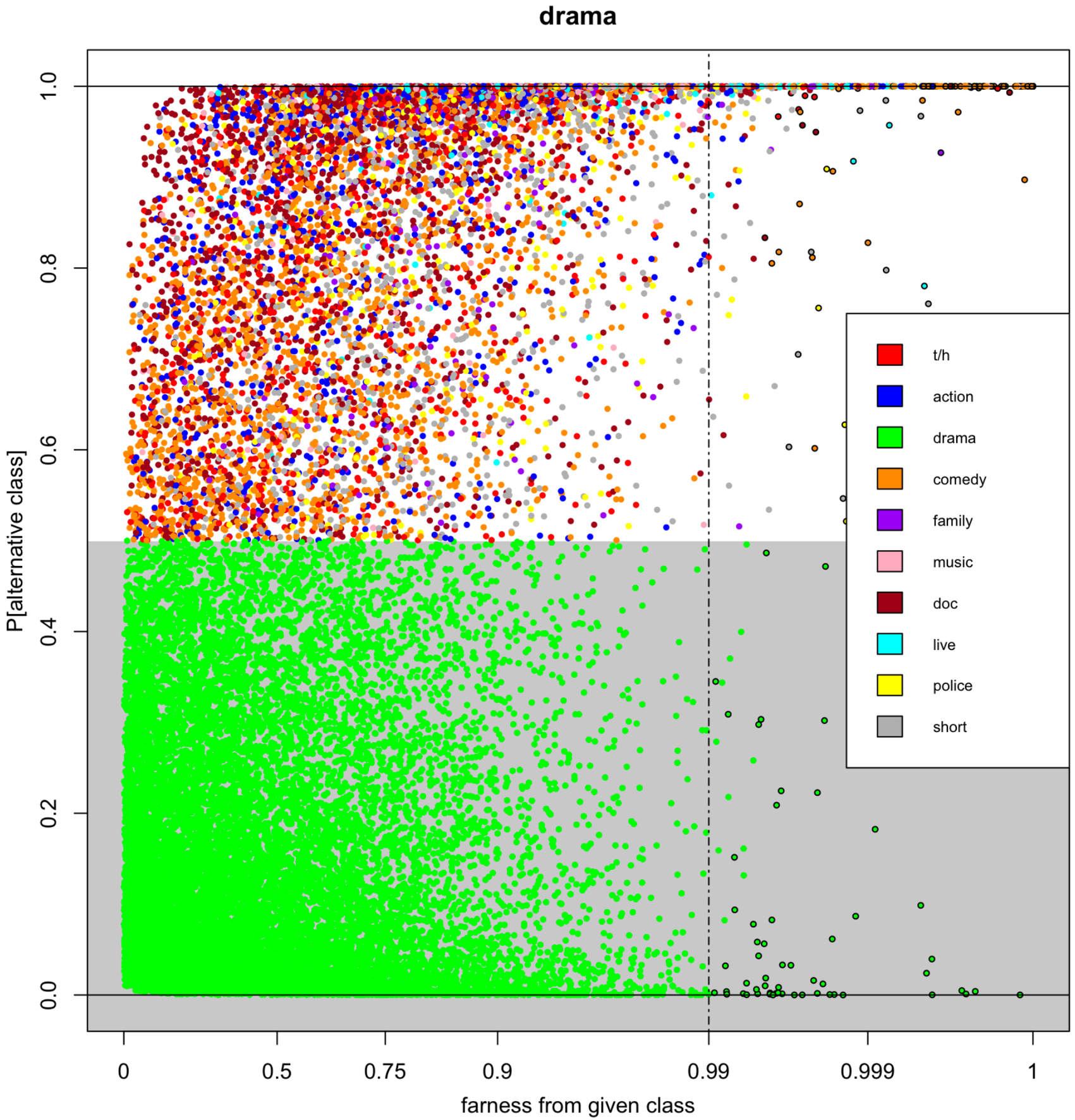
How it works

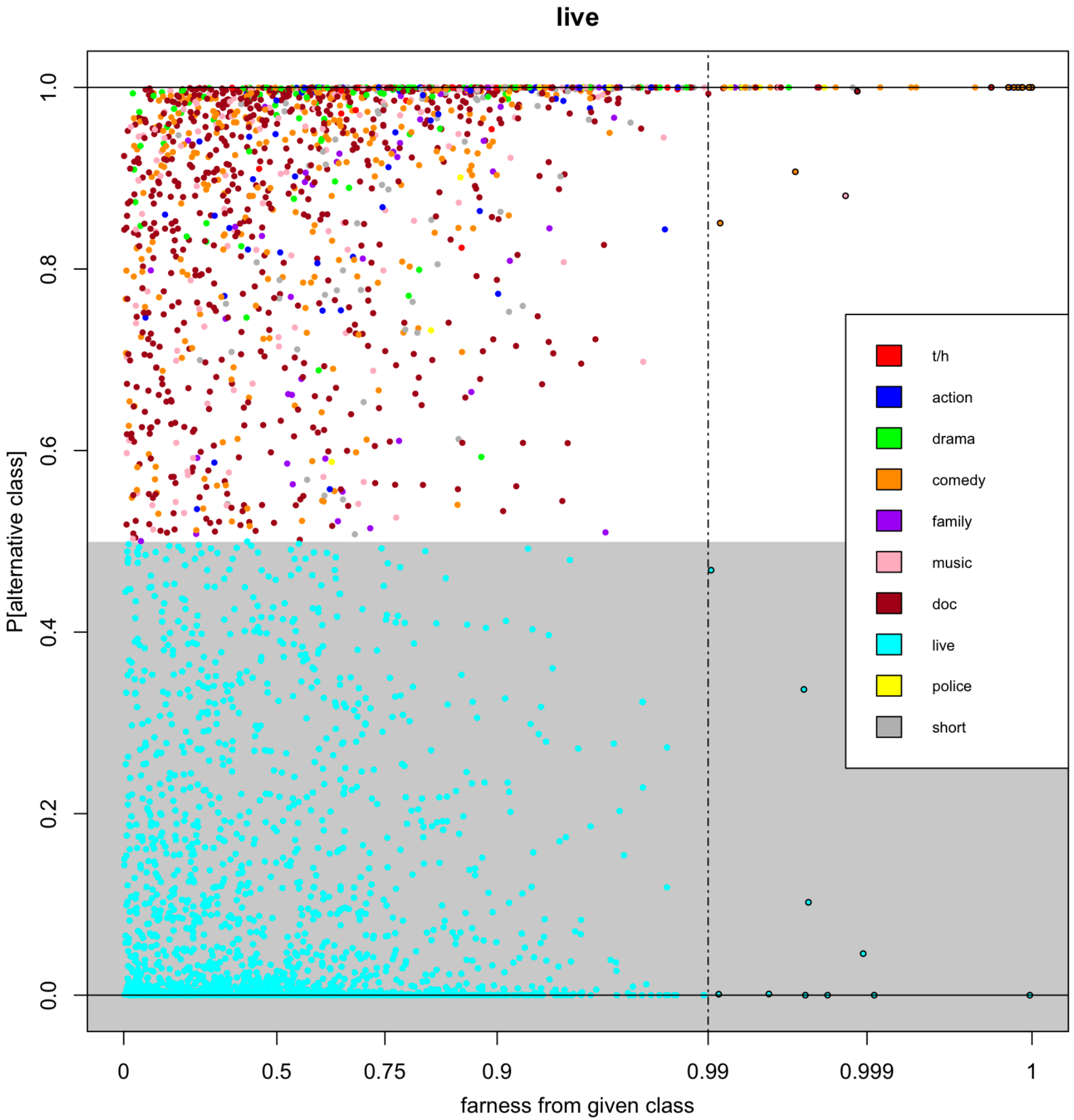
- assign object i to class $\operatorname{argmax}_{g=1,\ldots,G} \hat{p}(i,g)$
- object i has a known given label g_i
- $\tilde{p}(i) = \max\{\hat{p}(i,g)\,;\, g \neq g_i\}$
- $\mathrm{PAC}(i) = \dfrac{\tilde{p}(i)}{\hat{p}(i,g_i) + \tilde{p}(i)}$
- $\mathrm{farness}(i) = P[D(x,g_i) \leqslant D(i,g_i)]$
- $O(i) = \min\limits_{g=1,\ldots,G} \mathrm{farness}(i,g)$

How it works




thriller/horror
P[alternative class]
1.0
0.8
0.6
0.4
0.2
0.0
0
0.5
0.75
0.9
0.99
0.999
1
farness from given class
t/h
action
drama
comedy
family
music
doc
live
police
short


drama
P[alternative class]
1.0
0.8
0.6
0.4
0.2
0.0
0
0.5
0.75
0.9
0.99
0.999
1
farness from given class
t/h
action
drama
comedy
family
music
doc
live
police
short


live
P[alternative class]
1.0
0.8
0.6
0.4
0.2
0.0
0
0.5
0.75
0.9
0.99
0.999
1
farness from given class
t/h
action
drama
comedy
family
music
doc
live
police
short

Conclusion

Conclusion

—> Our goal is achieved: good accuracy but could be better

—> LDA performs better than CART

—> PCA improves the results

Further Improvements

—> Balancing the class distribution

—> Use more appropriate methods (SVM, Neural Network, etc)

To be continued…