https://github.com/RobertoDelGiudice/MovieGenresClassification

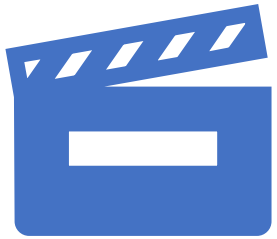
Movies Genres Classification

Big Data Analytics Project

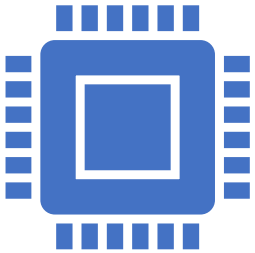
Made by **Roberto Del Giudice**
(Serial Num: 592998)

My objective

My project aims to apply a **Multi-Class Classification** model to a movie dataset.



The main goal is predict the genre of a single movie just by its own synopsis.



How will I do?

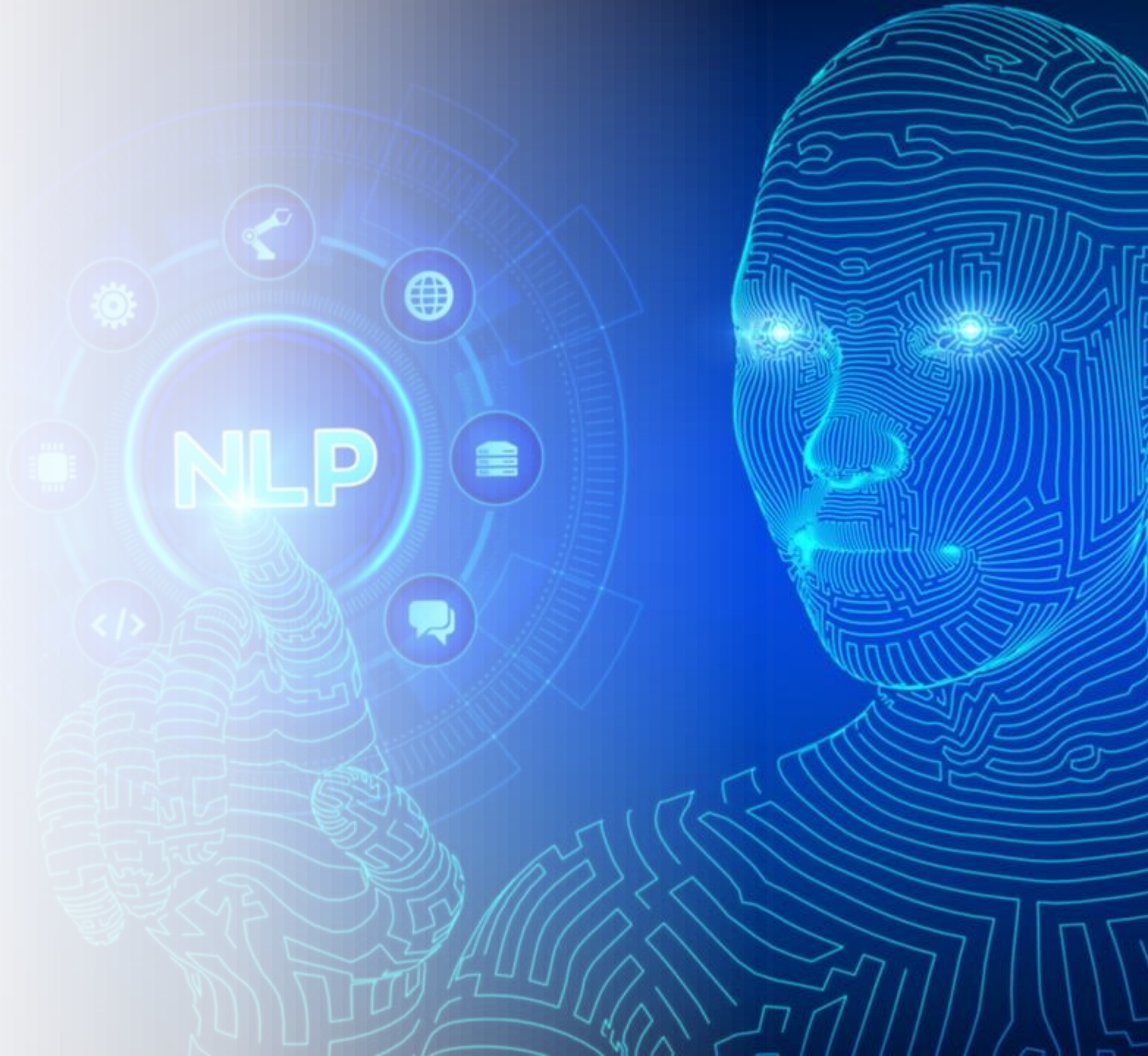
I've trained an Artificial Neural Network on an IMDB dataset, which provides Synopsis (as input X) and Genres (as output Y).

Why is it important?

- This process could be relevant for automatically updating or validating the genre of new movies (e.g. in an online catalogue).
- Also to suggest users, which movie might be perfect to watch based on its «*Genre Score*».

What do other works?

At the moment, there isn't a highly accurate classification model that can predict the genre fitting well based on a movie's title. Sure, you can consult a GPT to foresee it, but that is more or less what my project my project aims to accomplish with this NLP application.



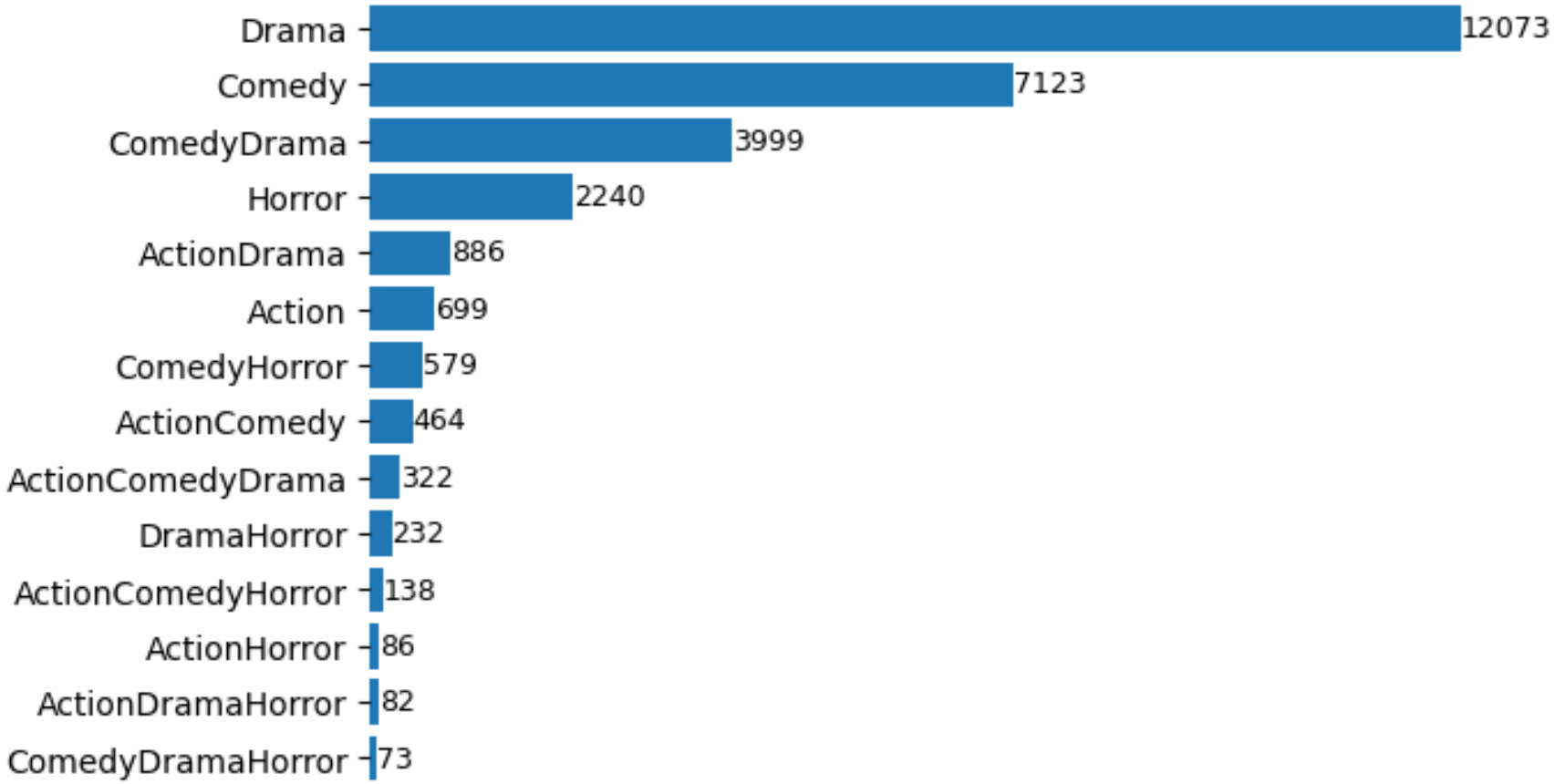
Before starting with enbeddings, let's Tidy our dataset

	imdb_title_id	title	genre	description
0	tt0000009	Miss Jerry	Romance	The adventures of a female reporter in the 1890s.
1	tt0000574	The Story of the Kelly Gang	Biography, Crime, Drama	True story of notorious Australian outlaw Ned ...
2	tt0001892	Den sorte drøm	Drama	Two men of high rank are both wooing the beaut...
3	tt0002101	Cleopatra	Drama, History	The fabled queen of Egypt's affair with Roman ...
4	tt0002130	L'Inferno	Adventure, Drama, Fantasy	Loosely adapted from Dante's Divine Comedy and...
...	...	...	...	...
85848	tt9905462	Pengalila	Drama	An unusual bond between a sixty year old Dalit...
85849	tt9906644	Manoharam	Comedy, Drama	Manoharan is a poster artist struggling to fin...
85850	tt9908390	Le lion	Comedy	A psychiatric hospital patient pretends to be ...
85851	tt9911196	De Beentjes van Sint-Hildegard	Comedy, Drama	A middle-aged veterinary surgeon believes his ...
85854	tt9914942	La vida sense la Sara Amat	Drama	Pep, a 13-year-old boy, is in love with a girl...

	imdb_title_id	title	description	Action	Comedy	Drama	Horror	Genre
0	tt0001892	Den sorte drøm	Two men of high rank are both wooing the beaut...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
1	tt0002461	Richard III	Richard of Gloucester uses manipulation and mu...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
2	tt0002646	Atlantis	After Dr. Friedrich's wife becomes mentally un...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
3	tt0003014	Il calvario di una madre	Single mother is separated from her children d...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
4	tt0003102	Ma l'amor mio non muore...	Leslie Swayne, an adventurer, in order to obta...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
...	...	...	...	...	...	...	...	...
28991	tt9905462	Pengalila	An unusual bond between a sixty year old Dalit...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]
28992	tt9906644	Manoharam	Manoharan is a poster artist struggling to fin...	0.0	1.0	1.0	0.0	[0.0, 1.0, 1.0, 0.0]
28993	tt9908390	Le lion	A psychiatric hospital patient pretends to be ...	0.0	1.0	0.0	0.0	[0.0, 1.0, 0.0, 0.0]
28994	tt9911196	De Beentjes van Sint-Hildegard	A middle-aged veterinary surgeon believes his ...	0.0	1.0	1.0	0.0	[0.0, 1.0, 1.0, 0.0]
28995	tt9914942	La vida sense la Sara Amat	Pep, a 13-year-old boy, is in love with a girl...	0.0	0.0	1.0	0.0	[0.0, 0.0, 1.0, 0.0]

I've selected movies that contain only genre from this list: ['***Action***', '***Comedy***', '***Drama***', '***Horror***']. Then I dropped the other ones.

The final dataset is compound by these movies:



The ***total*** is **28996** distinct movies

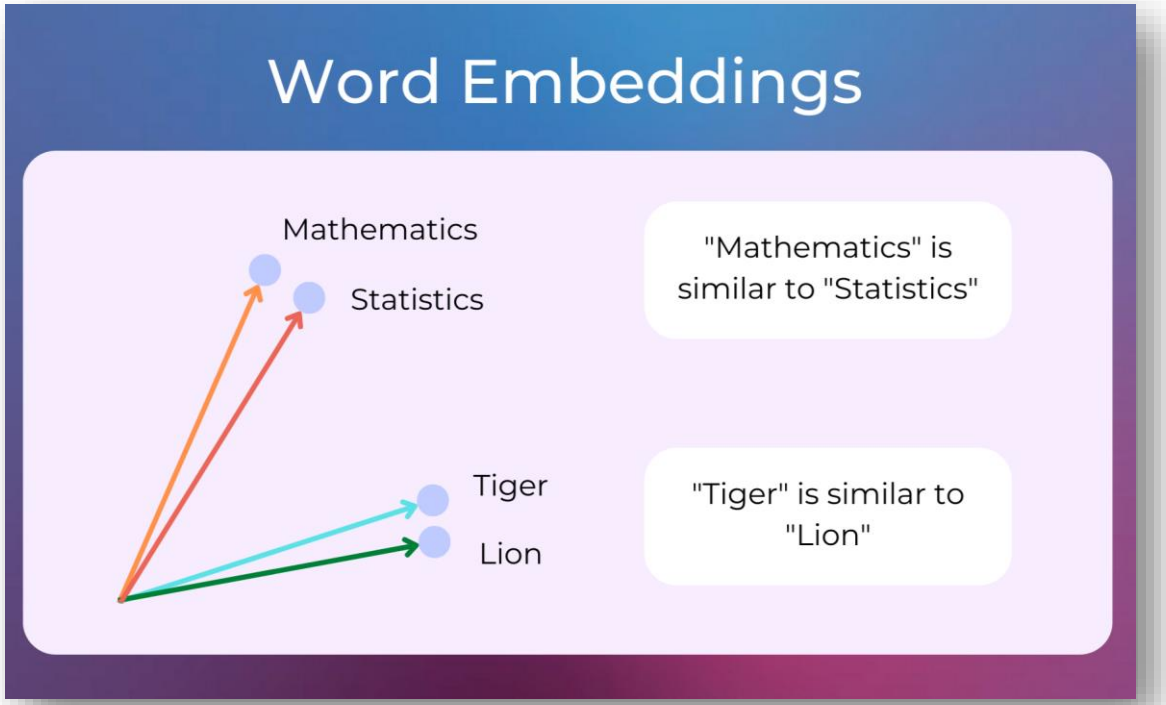
What embeddings is?

We cannot use input data as Natural Language, we must encode them into embeddings, which are numerical representation of the semantic and lexical meaning of a sentence for ML applications. I'll use BERT model, to obtain this conversion.

```
'The famous detective is pulled away
from retirement and his fiancée when the
condemned Moriarty escapes from prison
and swears vengeance.'
```

This synopsis becomes:

```
array([-2.03396827e-01, -1.08847693e-01,
3.33505034e-01, -8.47090334e-02 ...
-1.57784790e-01, -4.68660668e-02,
5.69955930e-02, -1.50748715e-01],
dtype=float32)
```
with shape `(768,)`

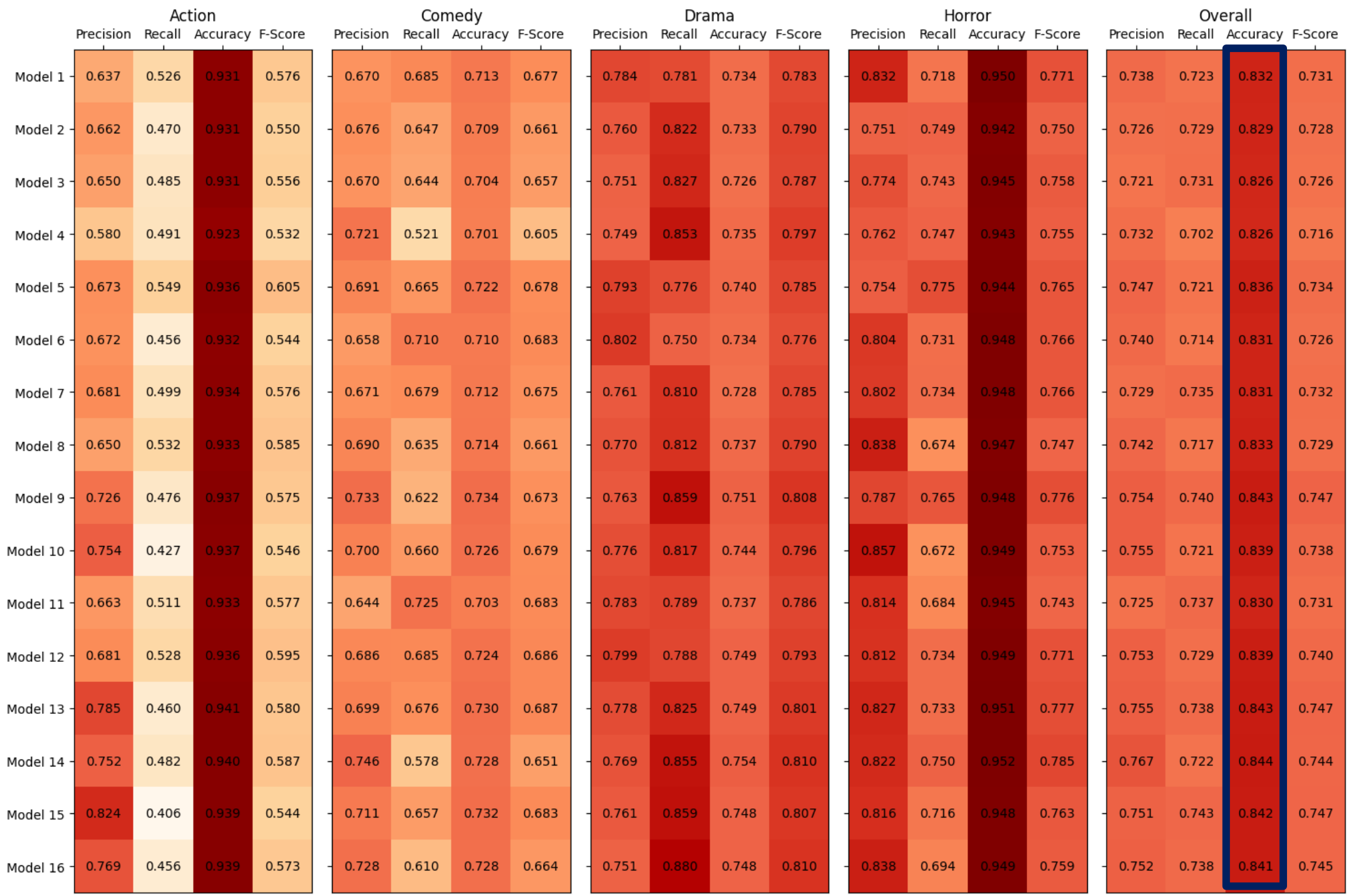


NEURAL NETWORK

In this phase, we need to choose the best configuration of the Neural Network to achieve the optimal performance on a *cross-validation dataset*. We have to tune two hyperparameters: the «**number of hidden layers**» and the «**number of neurons**».

I've tested 16 possible configurations, where each of them has 768 neurons in the input layer and only 4 neurons in the last output layers.

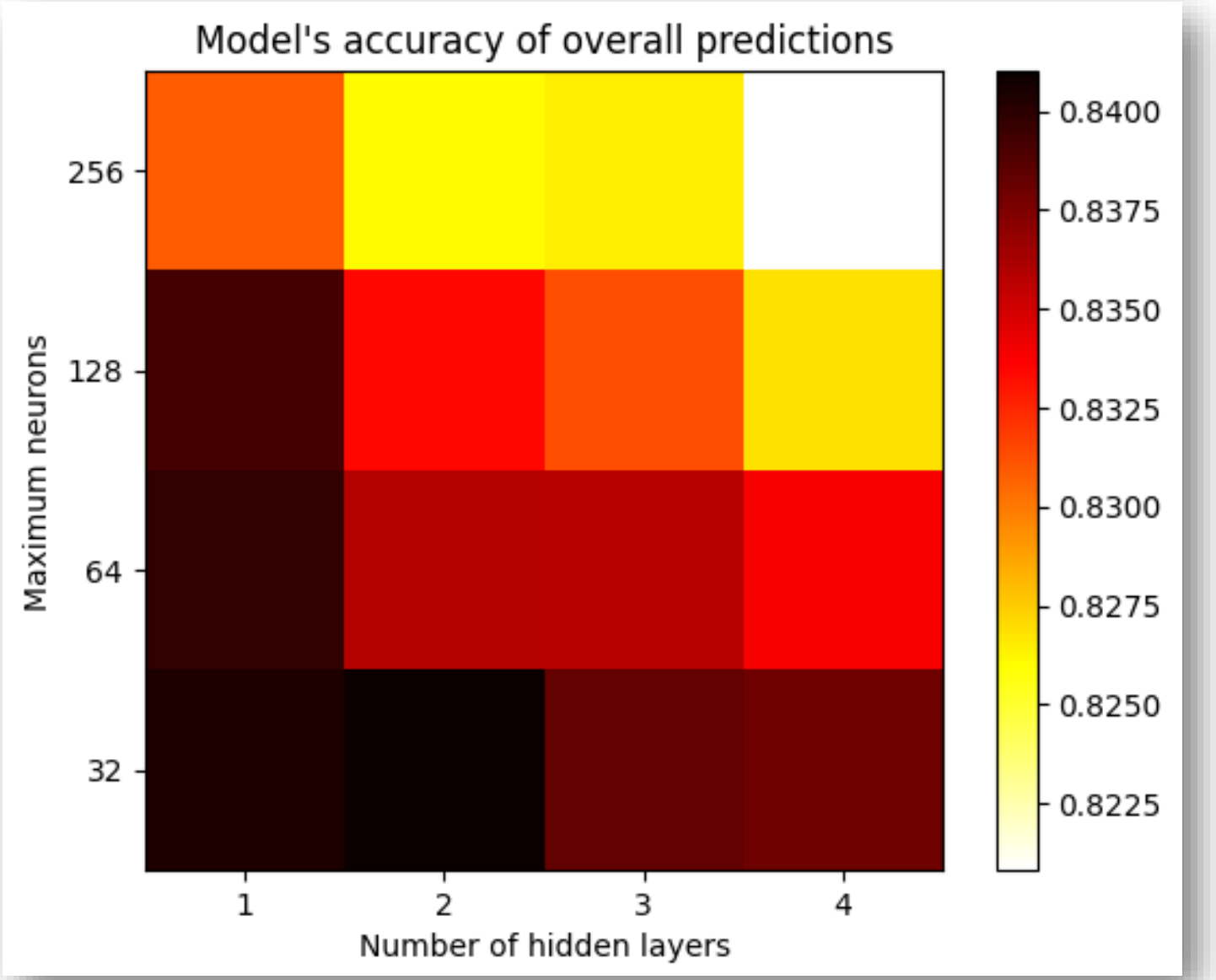
The hidden layers have «ReLu» as the activation function, while the output layer has «Sigmoid» as the activation function because we are dealing with a classification problem.


Action
Comedy
Drama
Horror
Overall
Precision
Recall
Accuracy
F-Score
Model 1 0.637 0.526 0.931 0.576 0.670 0.685 0.713 0.677 0.784 0.781 0.734 0.783 0.832 0.718 0.950 0.771 0.738 0.723 0.832 0.731
Model 2 0.662 0.470 0.931 0.550 0.676 0.647 0.709 0.661 0.760 0.822 0.733 0.790 0.751 0.749 0.942 0.750 0.726 0.729 0.829 0.728
Model 3 0.650 0.485 0.931 0.556 0.670 0.644 0.704 0.657 0.751 0.827 0.726 0.787 0.774 0.743 0.945 0.758 0.721 0.731 0.826 0.726
Model 4 0.580 0.491 0.923 0.532 0.721 0.521 0.701 0.605 0.749 0.853 0.735 0.797 0.762 0.747 0.943 0.755 0.732 0.702 0.826 0.716
Model 5 0.673 0.549 0.936 0.605 0.691 0.665 0.722 0.678 0.793 0.776 0.740 0.785 0.754 0.775 0.944 0.765 0.747 0.721 0.836 0.734
Model 6 0.672 0.456 0.932 0.544 0.658 0.710 0.710 0.683 0.802 0.750 0.734 0.776 0.804 0.731 0.948 0.766 0.740 0.714 0.831 0.726
Model 7 0.681 0.499 0.934 0.576 0.671 0.679 0.712 0.675 0.761 0.810 0.728 0.785 0.802 0.734 0.948 0.766 0.729 0.735 0.831 0.732
Model 8 0.650 0.532 0.933 0.585 0.690 0.635 0.714 0.661 0.770 0.812 0.737 0.790 0.838 0.674 0.947 0.747 0.742 0.717 0.833 0.729
Model 9 0.726 0.476 0.937 0.575 0.733 0.622 0.734 0.673 0.763 0.859 0.751 0.808 0.787 0.765 0.948 0.776 0.754 0.740 0.843 0.747
Model 10 0.754 0.427 0.937 0.546 0.700 0.660 0.726 0.679 0.776 0.817 0.744 0.796 0.857 0.672 0.949 0.753 0.755 0.721 0.839 0.738
Model 11 0.663 0.511 0.933 0.577 0.644 0.725 0.703 0.683 0.783 0.789 0.737 0.786 0.814 0.684 0.945 0.743 0.725 0.737 0.830 0.731
Model 12 0.681 0.528 0.936 0.595 0.686 0.685 0.724 0.686 0.799 0.788 0.749 0.793 0.812 0.734 0.949 0.771 0.753 0.729 0.839 0.740
Model 13 0.785 0.460 0.941 0.580 0.699 0.676 0.730 0.687 0.778 0.825 0.749 0.801 0.827 0.733 0.951 0.777 0.755 0.738 0.843 0.747
Model 14 0.752 0.482 0.940 0.587 0.746 0.578 0.728 0.651 0.769 0.855 0.754 0.810 0.822 0.750 0.952 0.785 0.767 0.722 0.844 0.744
Model 15 0.824 0.406 0.939 0.544 0.711 0.657 0.732 0.683 0.761 0.859 0.748 0.807 0.816 0.716 0.948 0.763 0.751 0.743 0.842 0.747
Model 16 0.769 0.456 0.939 0.573 0.728 0.610 0.728 0.664 0.751 0.880 0.748 0.810 0.838 0.694 0.949 0.759 0.752 0.738 0.841 0.745

Heatmap of overall accuracy (1)

To choose the best configuration, I considered the *accuracy** of all genres concatenated (precisely *overall*), calculated on the cross-validation dataset which comprises the 20% of the initial dataset.

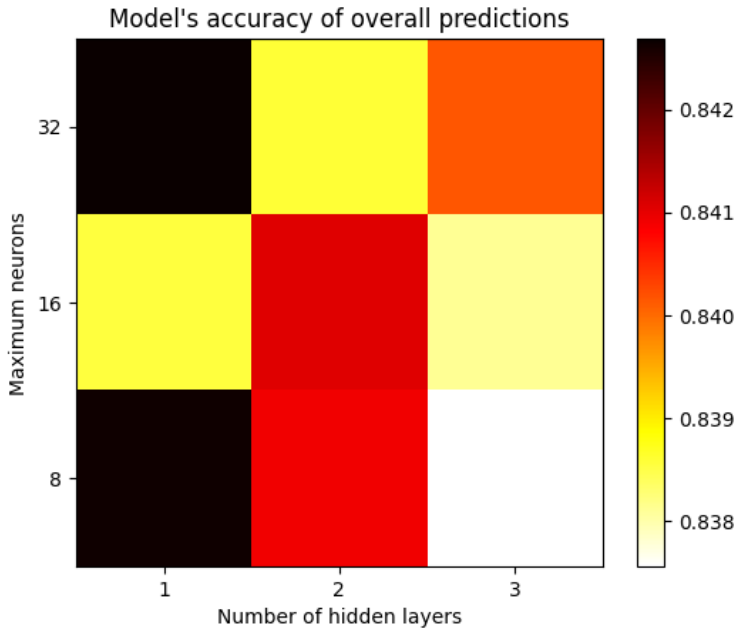
I noticed best performances are obtained when we have a low number of hidden layers and a low number of neurons.

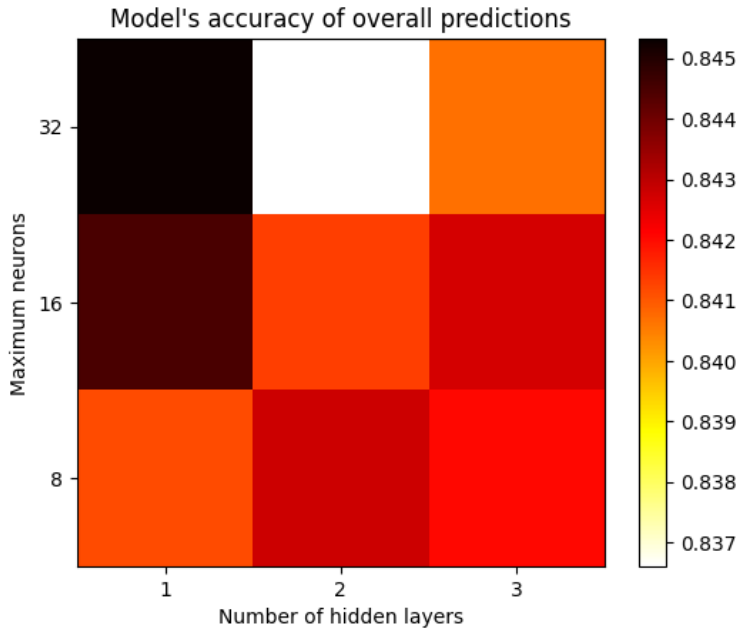


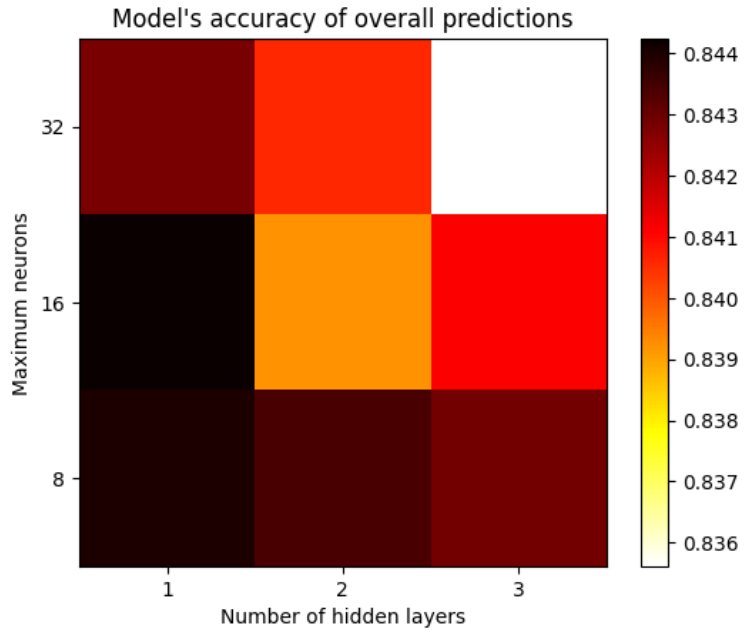
$$* Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Heatmap of overall accuracy (2)

So I decided to test other configurations with a lower number of neurons and a lower number of hidden layers. I've conducted three different simulations, and the outcomes suggest that one hidden layer is sufficient with approximately 16 neurons.



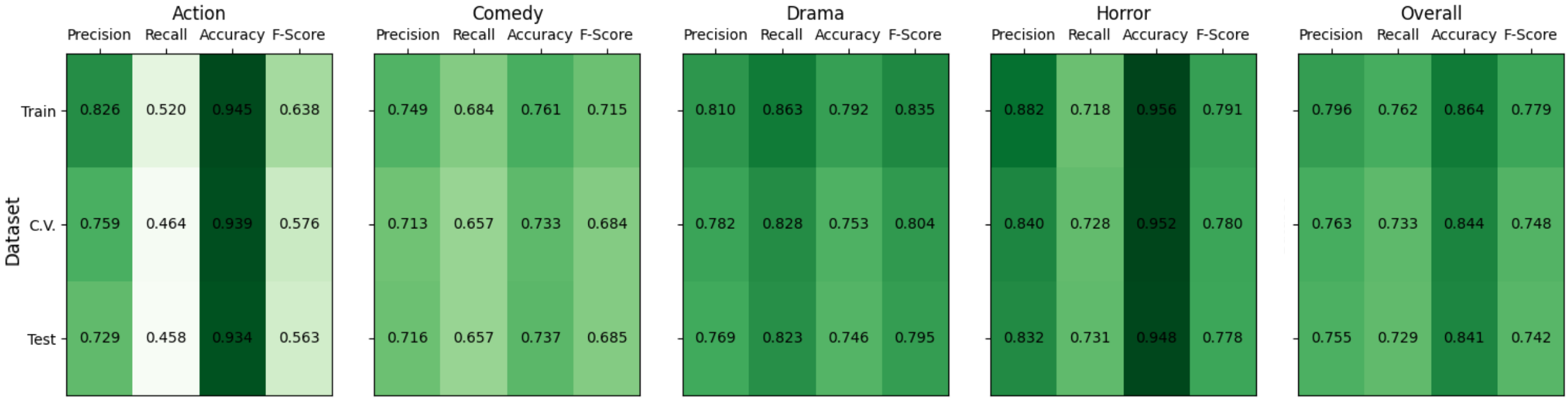




CONCLUSIONS

The best performance (highest overall accuracy on a testing dataset) is achieved when using only one hidden layer with 16 neurons. The results are visualized in the plot below.

Here, the three datasets have been generated by splitting the initial dataset into three parts, each comprising 60%, 20%, and 20% of the total data, respectively.




Thank you for the
attention