Machine Learning on Big Data
Task 1 and 2

Alexia GROSS

Machine Learning on Big Data
Department of Computer Science
Saint-Petersburg Electrotechnical University

08 Nov 2021

Overview

- Analytic task for selected data
  - General information of the selected data set
  - The data analysis target
  - Meta-information
  - Meta-information Summary
  - Data restrictions
- Data analysis by Big data machine learning tools
  - Machine learning algorithm for solving the task
  - Necessary data settings for the machine learning algorithm
  - Building time depending on the number of hosts
  - Building time on the number of data
  - Conclusions

Analytic task for selected data

General information of the selected dataset

- Dataset : `https://www.kaggle.com/c/tmdb-box-office-prediction/overview`
- Source : Kaggle
- Provided by : TMDB
- Analytic Tasks : Predict the overall worldwide box office revenue of movies

The data analysis target

Goal

To determine a movie's popularity before its release

Meta-information

- Format : Data stored in two CSV files :
  - train_set.csv [to train our model]
  - test_set.csv [to test our model]
- 22 attributes
- 3 Types : int64, float64 or object

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3000 entries, 0 to 2999
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   id                     3000 non-null   int64
 1   belongs_to_collection  604 non-null    object
 2   budget                 3000 non-null   int64
 3   genres                 2993 non-null   object
 4   homepage               946 non-null    object
 5   imdb_id                3000 non-null   object
 6   original_language      3000 non-null   object
 7   original_title         3000 non-null   object
 8   overview               2992 non-null   object
 9   popularity             3000 non-null   float64
 10  poster_path            2999 non-null   object
 11  production_companies   2844 non-null   object
 12  production_countries   2945 non-null   object
 13  release_date           3000 non-null   object
 14  runtime                2998 non-null   float64
 15  spoken_languages       2980 non-null   object
 16  status                 3000 non-null   object
 17  tagline                2403 non-null   object
 18  title                  3000 non-null   object
 19  Keywords               2724 non-null   object
 20  cast                   2987 non-null   object
 21  crew                   2984 non-null   object
 22  revenue                3000 non-null   int64
dtypes: float64(2), int64(3), object(18)
memory usage: 539.2+ KB
```

Figure: List of attributes

Data Restrictions

Missing Values

The dataset contains 3000 entries.
For each attribute, we have the number of "Non-Null" values.

Attributes	Types	Number of missing values
belongs_to_collection	object	2 396
budget	int64	0
genres	object	7
homepage	object	2 054
imdb_id	object	0
original_language	object	0
original_title	object	0
overview	object	8
popularity	float64	0
poster_path	object	1
production_companies	object	156
production_countries	object	65
release_date	object	0
runtime	float64	2
spoken_languages	object	20
status	object	0
tagline	object	597
title	object	0
Keywords	object	276
cast	object	13
crew	object	16
revenue	int64	0

Figure: Number of missing values

Necessary data settings for the machine learning algorithm

- The attribute "Revenue" is removed.
  - The goal is to predict a movie's popularity before its release
  - Absurd to include the revenue.
- All the attributes which contain less than 4000 non-null data are removed.
  - belongs_to_collection
  - homepage
  - tagline
- Multimodal variables need to be taken into account: use of pre-treatment system developped by Hoang Dang
  1. Use a predefined class "InfoExtractor" to separate and extract the data
  2. Use a predefined class "TextEncoder" to collect all the "strings" variables and convert them into "float" variables.
  3. The data can be used

Necessary data settings for the machine learning algorithm

	id	belongs_to_collection	budget	genres	homepage	imdb_id	original_language	original_title	overview	popularity
0	1	[{'id': 313576, 'name': 'Hot Tub Time Machine ...	14000000	[{'id': 35, 'name': 'Comedy'}]	NaN	tt2637294	en	Hot Tub Time Machine 2	When Lou, who has become the "father of the In...	6.5754
1	2	[{'id': 107674, 'name': 'The Princess Diaries ...	40000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	tt0368933	en	The Princess Diaries 2: Royal Engagement	Mia Thermopolis is now a college graduate and ...	8.2489

Figure: Step 1

	genres	production_companies	production_countries	spoken_languages	Keywords
0	Comedy	4 60 8411	US	English	time_travel sequel hot_tub duringcreditsstinger
1	Comedy Drama Family Romance	2	US	English	coronation duty marriage falling_in_love
2	Drama	2266 3172 32157	US	English	jazz obsession conservatory music_teacher new_...
3	Thriller Drama	NaN	IN	English हिन्दी	mystery bollywood police_corruption crime indi...
4	Action Thriller	NaN	KR	한국어/조선말	NaN

Figure: Step 2

Necessary data settings for the machine learning algorithm

	budget	genres	original_language	original_title	overview	poster_path	production_companies	production_countries	runtime	spoken_languages
0	14000000	1.3649	0.7722	29.7698	86.2111	8.0070	10.7776	0.8390	93.0000	0.7634
1	40000000	6.8065	0.7722	32.2026	232.2292	8.0070	3.8840	0.8390	113.0000	0.7634
2	3300000	1.0846	0.7722	7.3139	75.4125	8.0070	17.8654	0.8390	105.0000	0.7634
3	1200000	2.6527	4.2594	7.3139	318.5386	8.0070	0.0000	3.6266	122.0000	4.7455
4	0	3.1861	4.9688	7.3139	99.7664	8.0070	0.0000	4.8785	118.0000	8.7627

Figure: Step 3

Data analysis by Big data machine learning tools

Machine learning algorithm for solving the task

- Sequential model used which is more appropriate for a pile of superficial dense layers.
- This model is multimodal because we have different types of attributes : integer or float
- Keras from TensorFlow was used to perform this experiment

Experiment 1 - Details

- Neural network composed of an alternative of dense layers and dropout layers
- The dropout layers permit to avoid the overfitting on the learning data
- The dense layers connect each neuron of the defined layer to the neuron of the previous layer.
- The loss is "MSE" (Mean squared error) for a quadratic error, and the optimizer used is "Adam."
- 30 epochs and three layers, which; the last one is a linear activation function.

Experiment 1 - Results

```
Epoch 100/100
75/75 [==============================] - 0s 4ms/step - loss: 24.8019 - val_loss: 21.7725
```

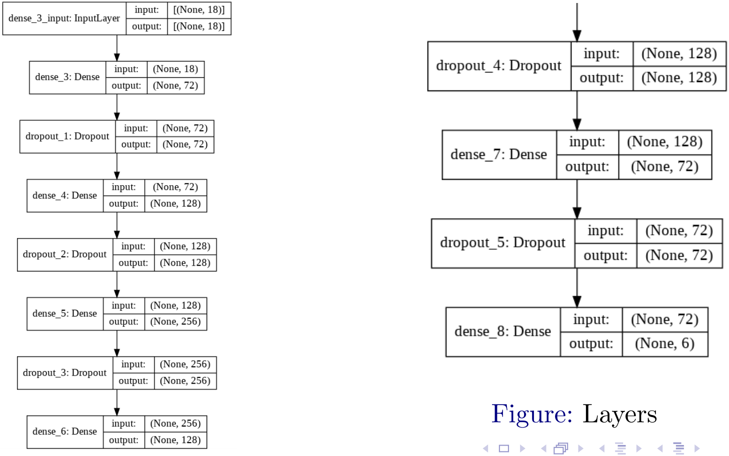


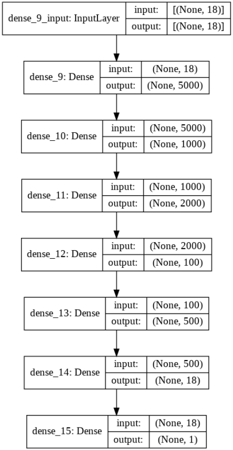
Figure: Layers

Experiment 2 - Details

- Seven dense layers used with, for each, an activation function ReLu
- The model is compiled with "mean square logarithmic error" as the loss function and with "SGD" (Stochas- tic gradient descent) as the optimizer.
- The model is trained on 100 epochs and allows us to obtain a loss a value around 0.23 !

Experiment 2 - Results

```
Epoch 100/100
75/75 [==============================] - 0s 6ms/step - loss: 0.2316 - val_loss: 0.3382
```



Comparison

Experiment 1
6 dense layers + 5 dropout layers
100 epochs
Loss : mse
Optimizer : adam
Result : val_loss = 20

Experiment 2
7 dense layers
100 epochs
Loss : mean square logarithmic
Optimizer : sgd
Result : val_loss = 0,2

Movies' popularity predicted

	id	original_title	popularity	popularity_predicted
0	3001	ディアルガVSパルキアVSダークライ	3.8515	2.8183
1	3002	Attack of the 50 Foot Woman	3.5598	4.2085
2	3003	Addicted to Love	8.0852	2.6045
3	3004	Incendies	8.5960	7.6022
4	3005	Inside Deep Throat	3.2177	0.5571
5	3006	SubUrbia	8.6793	4.0514
6	3007	Drei	4.8989	5.4300
7	3008	The Tigger Movie	7.0234	5.8173
8	3009	Becoming Jane	7.8297	6.7551
9	3010	Toy Story 2	17.5477	16.5054
10	3011	Cruel World	0.2624	3.5962
11	3012	Bande de filles	4.2203	4.5207
12	3013	The Gods Must Be Crazy	10.9735	5.7875
13	3014	Raising Victor Vargas	1.1787	1.7933
14	3015	The Brothers Bloom	7.9731	6.8127
15	3016	Beautiful Boy	2.1148	4.3460
16	3017	Hot Tub Time Machine	11.9677	7.0565
17	3018	Transcendence	9.7302	10.0890
18	3019	All That Jazz	5.6323	8.1964

Building time depending on the number of hosts

Parameters added for multiworkers :

- tf.distribute.MultiWorkerMirroredStrategy : implements a synchronous CPU/GPU multi-worker solution
- workers: Integer.
- use_multiprocessing: Boolean. Used for generator

	Single worker	Multiprocessing
Description	Same as before	I added several parameters as describe above
Function used	`model_3.fit( X_train_pp, y_train, epochs=100, batch_size=32, validation_split=0.2, callbacks=[cb])`	`model_3.fit( X_train_pp, y_train, epochs=100, batch_size=32, validation_split=0.2, callbacks=[cb], workers=8, use_multiprocessing=True)`
Time for training	54.683 s	130 s

Figure: Comparison

Building time depending on the number of hosts

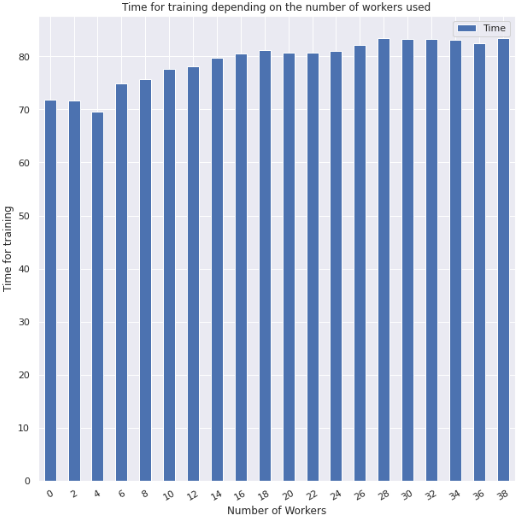


Figure: Time for training depending on the number of workers used

Building time depending on the number of hosts

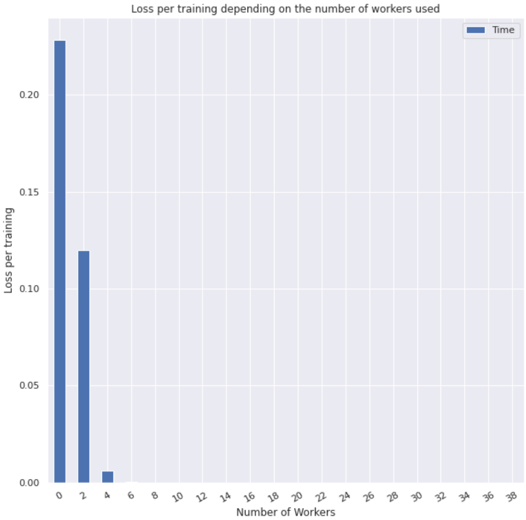


Figure: Loss per training depending on the number of workers used

Building time depending on the number of data

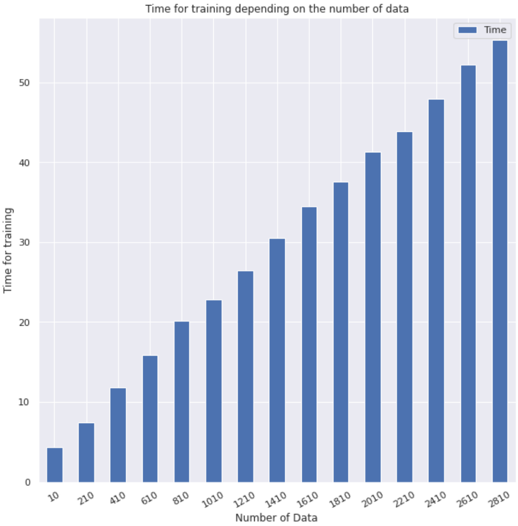


Figure: Time for training depending on the number of data used

Building time depending on the number of data

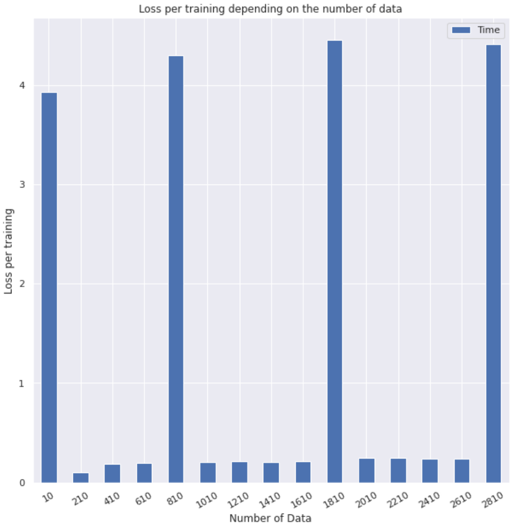


Figure: Loss per training depending on the number of data used

Building time depending on the number of data

Originally, we have 3000 data. If we reduced it to 1500 data : the training time dropped from 54.683 s to 30 s.
But the prediction accuracy dropped as well.

	id	original_title	popularity	popularity_predicted
0	3001	ディアルガVSパルキアVSダークライ	3.8515	0.3077
1	3002	Attack of the 50 Foot Woman	3.5598	0.3375
2	3003	Addicted to Love	8.0852	0.3725
3	3004	Incendies	8.5960	0.3878
4	3005	Inside Deep Throat	3.2177	0.3013
5	3006	SubUrbia	8.6793	0.3494
6	3007	Drei	4.8989	0.3425
7	3008	The Tigger Movie	7.0234	0.3617
8	3009	Becoming Jane	7.8297	0.4029
9	3010	Toy Story 2	17.5477	4.0935
10	3011	Cruel World	0.2624	0.3098
11	3012	Bande de filles	4.2203	0.5442
12	3013	The Gods Must Be Crazy	10.9735	0.3664
13	3014	Raising Victor Vargas	1.1787	0.3319
14	3015	The Brothers Bloom	7.9731	0.3606

Conclusion

The experiments carried out have been conclusive.
Nevertheless, in the end, I cannot predict if a movie will be able to reimburse the cost of its production or if it will be the next icon of pop culture.
To interpret and use these results, we could perhaps predict the probability of a movie being the next prominent movie.