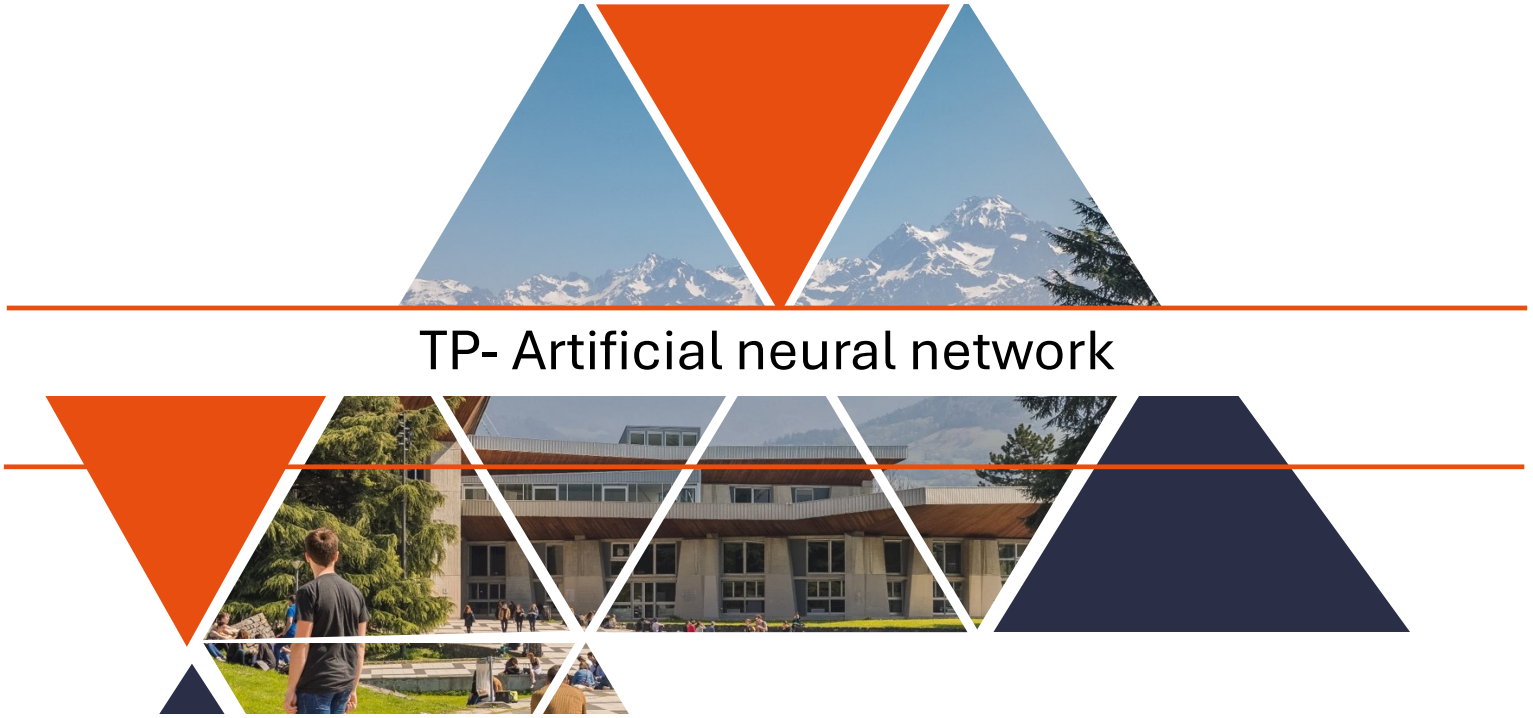

TP- Artificial neural network

Parcours Progis

Etudes, Medias, communication, Marketing

Bahareh Afshinpour

01.07.2026

Université Grenoble Alpes

Sciences Po Grenoble UGA

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

TensorFlow is an open-source library developed by Google for Machine Learning and Deep Learning.
It provides tools for creating, training, and evaluating Artificial Neural Networks.
TensorFlow automatically performs all the complex mathematical calculations required to train the network.

Python

```
%pip install tensorflow
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Dataset

```
df = pd.read_csv("social_media_engagement_dataset.csv")
df.head()
```

	platform	post_type	sentiment	weekday	hour	caption_length
0	Instagram	Carousel	Neutral	Thu	22	269
1	LinkedIn	Image	Neutral	Mon	10	39
2	TikTok	Video	Positive	Sat	11	75
3	LinkedIn	Video	Negative	Thu	11	261
4	Facebook	Carousel	Positive	Mon	13	277

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Explore the dataset

```
df.shape
```

```
(600, 19)
```

```
df['high_engagement'].value_counts()
```

```
high_engagement
0    324
1    276
Name: count, dtype: int64
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Separate input and output (Target variable)

```
X = df.drop("high_engagement", axis=1)
y = df["high_engagement"]
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Convert categorical variables

Categorical variables are variables that contain **text** labels rather than **numerical** values.

```
X = pd.get_dummies(X, drop_first=True)
```

```
X.head()
```

	hour	caption_length	hashtag_count	follower_count
0	22	269	6	5849
1	10	39	3	6007
2	11	75	7	500
3	11	261	5	6317
4	13	277	4	6460

5 rows × 28 columns

In this dataset, the categorical variables are:

- Platform (Instagram, Facebook, TikTok, LinkedIn)
- post_type(Image, Video, Carousel, Text)
- Sentiment (Positive, Neutral, Negative)
- Weekday

These variables must be converted into numerical values before training the neural network.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Split train and test data

Why should we not train and test the model on the same data?

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)
```

If we train and test the model on the same data, the model may simply memorize the training examples instead of learning general patterns.

Using separate training and test datasets allows us to evaluate how well the model generalizes to future social media posts.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Scale the data

Neural networks perform better when numerical variables are on a similar scale. Scaling helps the model learn more efficiently and converge faster.

follower_count may contain values in the **thousands**.
hashtag_count may contain values between **0 and 20**.

```
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

The variables in the dataset may have very different ranges.
Scaling places all variables on a similar scale, allowing the neural network to learn more efficiently and converge faster.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Create the neural network

- Artificial Neural Networks are composed of layers of interconnected neurons. In this project, we will build a simple neural network composed of:
  - One input layer
  - Two hidden layers
  - One output layer

❖ A first hidden layer with 16 neurons and ReLU activation.
❖ A second hidden layer with 8 neurons and ReLU activation.
❖ An output layer with 1 neuron and Sigmoid activation.

```
model = Sequential()

model.add(Dense(16, activation='relu', input_shape=(X_train.shape[1],)))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

The Sigmoid function produces an output between 0 and 1. Sigmoid is commonly used in the output layer of binary classification models.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Create the neural network

```
model.summary()
```

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 16)	464
dense_7 (Dense)	(None, 8)	136
dense_8 (Dense)	(None, 1)	9

Total params: 1,829 (7.15 KB)
Trainable params: 609 (2.38 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 1,220 (4.77 KB)

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Compile the Model

- Compile the model using: Optimizer: Adam, Loss function: Binary Crossentropy, Metric: Accuracy

```
model.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy']
)
```

Before training, the neural network must know:

- How to update its weights.
- How to measure prediction errors.
- How to evaluate its performance.

The compile step configures these settings and prepares the model for training.
Without compiling, the neural network cannot start the learning process.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Train the Model

- An epoch corresponds to one complete pass through the entire training dataset.
- During each epoch:
  - The neural network makes predictions.
  - The loss is calculated.
  - The error is propagated backward through the network.
  - The weights are adjusted.
  - The model improves its predictions.

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Train the Model

- During training, the neural network learns by adjusting its weights to minimize prediction errors.
- Train the model for 50 epochs.Use a batch size of 16.Reserve 20% of the training data for validation.

```
history = model.fit(
    X_train,
    y_train,
    epochs=50,
    batch_size=16,
    validation_split=0.2
)
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Evaluate the Model

- After training, we evaluate the neural network on data that it has never seen before.

```
loss, accuracy = model.evaluate(X_test, y_test)

print("Test Accuracy:", accuracy)
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8383 - loss: 0.3723
Test Accuracy: 0.8666666746139526
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Make Predictions

- The trained neural network can now predict whether a new post is likely to generate high engagement.

Examples:
•0.95 → very likely to have high engagement
•0.75 → likely to have high engagement
•0.40 → likely to have low engagement
•0.05 → very unlikely to have high engagement

```
y_pred_prob = model.predict(X_test)
y_pred = (y_pred_prob > 0.5).astype(int)
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Confusion matrix et Classification Report

- A confusion matrix helps us understand the types of errors made by the model.
- The classification report provides detailed performance metrics.

```
confusion_matrix(y_test, y_pred)
```

```
array([[62,  5],
       [11, 42]])
```

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.85      0.93      0.89        67
           1       0.89      0.79      0.84        53

    accuracy                           0.87       120
   macro avg       0.87      0.86      0.86       120
weighted avg       0.87      0.87      0.87       120
```

PROGIS
ETUDES, MEDIAS, COMMUNICATION, MARKETING

Confusion matrix et Classification Report

- **Accuracy**
  - Measures the overall percentage of correct predictions.
  - Useful when the classes are balanced.

- **Precision**
  - Measures how many posts predicted as High Engagement were actually High Engagement.
  - High precision means few false positives.
  - Precision is important when marketing resources are limited and we want to avoid promoting posts that are unlikely to perform well.

- **Recall**
  - Measures how many truly High Engagement posts were correctly identified.
  - High recall means few false negatives.
  - Recall is important when we want to identify as many successful posts as possible.

- In this project, Recall can be considered the most important metric :
  - because marketers generally want to identify as many potentially successful posts as possible.
  - Missing a post that could generate high engagement may represent a lost opportunity.
  - Therefore, a model with high recall helps detect a larger number of high-performing posts.

END