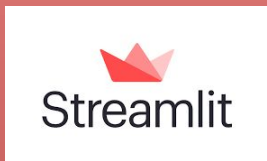


# Développer une application web Data / ML avec **Streamlit.io**



Formation complète

Projet étudié :



Sélection et Analyse du cours d'une  
action du S&P 500



Formateur :



[Pierre-Louis Danieau](#),  
Ingénieur Data Scientist



# Let's start !



pl.danieau@gmail.com

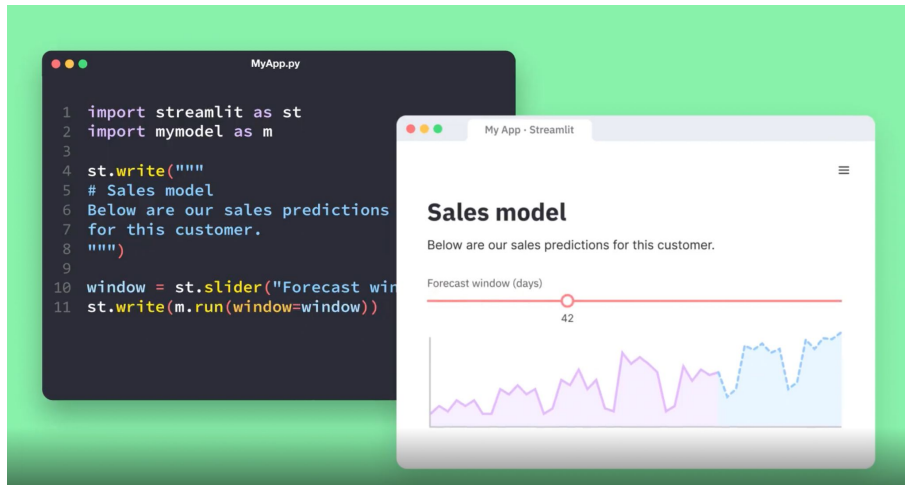


Pierre-Louis Danieau,  
Ingénieur Data Scientist



# Qu'est ce que Streamlit ?

- ❖ Streamlit est un Framework Python Open Source.
- ❖ Streamlit permet de créer une application web avec uniquement la maîtrise de Python.



[Lien de la vidéo de présentation](#)

## Pré requis :

- ❖ La connaissance du langage Python (Indispensable).

# A quoi sert Streamlit ?

## 3 cas d'usage principaux :

### 1 Partager ses travaux au monde entier

#### Welcome To Project Eagle Vision!

Either upload your own image or select from the gallery to get a pre-trained model. The image you select or upload will be fed through the Deep Neural Network to recognize and the output will be displayed to the screen.

Upload An Image

Drag and drop file here

or click to select from files

Browse Files

#### Here is the image you've selected



#### Here are the five most likely bird species

	Species	Confidence Level
1	African Crested Crane	95.940%
2	Ostrich	9.823%
3	Marabou Stork	0.990%
4	Flamingo	0.000%
5	Emu	0.000%

@joshmantova

Tu as créé un modèle de classification d'image ? Laisse le monde l'utiliser !

### 2 Développer un MVP (Minimum Viable Product)

#### Real Time Speech-to-Text

This demo app is using [DeepSpeech](#), an open speech-to-text engine.

A pre-trained model released with [v0.9.3](#), trained on American English is being served.

Choose the app mode

Sound only (sendonly)

START

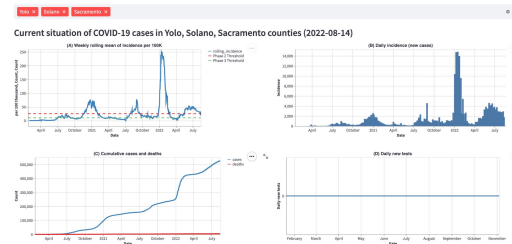
SELECT DEVICE

@Yuichiro

Une idée de projet ? Crée un MVP pour valider ton idée !

### 3 Utilisation par des Data Scientists en entreprise

Select counties of interest



@Pranav-Pandit

Construit et partage des dashboard interactifs !

# Les étapes d'un projet Data

Différents moyens de déployer son projet Data / ML :

Etape 1 : Exploration

jupyter

Data

Modèle



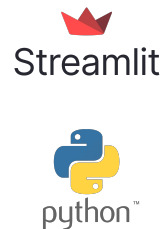
Production  
(sans Streamlit)



Difficile

Etape 2 : Déploiement




Production  
(avec Streamlit)



Simple

Streamlit

Comparaison des Frameworks Python de développement web :

			
Language	Python, HTML, CSS...	Python, HTML, CSS...	Python
Prise en main	Difficile	Modéré	Facile
Déploiement	AWS, AZUR, GCP...	AWS, AZUR, GCP...	Streamlit cloud
Personnalisable	Hautement	Modéré ++	Modéré -
Communauté	Grande	Grande	Grande

## 1. Préparer son environnement de travail

- a. Installer Streamlit / Github / Environnement d'exécution/

## 2. Les fondations de Streamlit

- a. Afficher du texte / photos / data ... – Exercices
- b. Création de la structure de l'application finale – Projet

## 3. Interagir avec l'utilisateur

- a. Création de widgets pour interagir avec l'internaute. – Exercices
- b. Création des widgets du projet – Projet

## 4. Les visualisations

- a. Afficher des visuels – Exercices
- b. Création des visuels du projet + Machine Learning – Projet

## 5. Les paramètres avancés

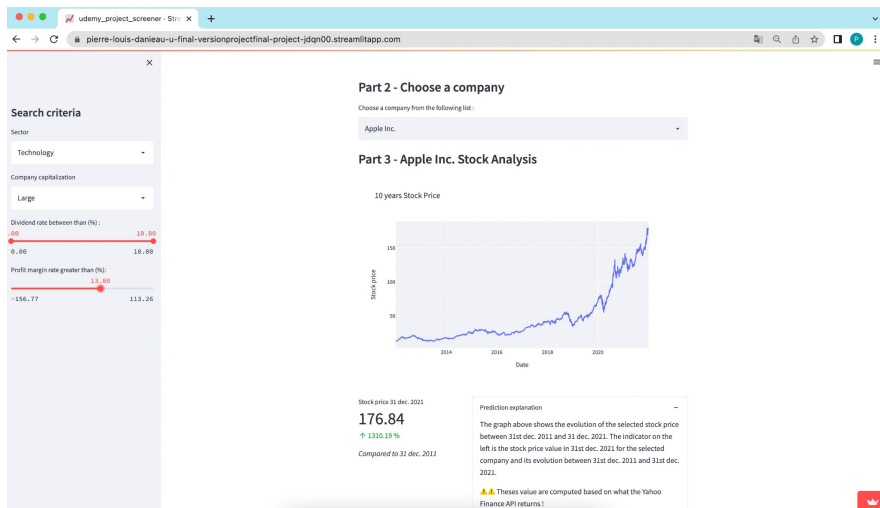
## 6. Le déploiement de l'application

- a. Rendre l'app accessible sur le web – Projet

Vous apprendrez à développer votre  
1ère application web avec **Streamlit**  
(Exercices +Projet)

## Application : S&P 500 Screener & Analysis

**Récolter** des données sur les 500 plus grosses entreprises américaines puis les **analyser** afin d'afficher l'**évolution du cours** des actions du S&P 500 ainsi que leurs performances !



### 1) Etape 1 :

Récolter et afficher des données sur les entreprises du S&P 500 (500 plus grosses capitalisations aux USA).

### 2) Etape 2 :

Sélection d'une entreprise et affichage de l'évolution du cours de son action sur 10 ans.

[Screener & Stock analysis demo](#)



# 1/ Préparation de son environnement de travail



## 1) Etape 1 : Installation de Python

- ❖ Installez python (si ce n'est pas déjà fait) : [Lien de téléchargement](#) (J'utilise la version 3.9.6 dans cette formation).

## 2) Etape 2 : Installation de votre IDE favori (VScode, Pycharm, Spyder...)

- ❖ Installez votre IDE favori pour commencer à coder (Aucune préférence). Dans le tutoriel j'utilise [VScode](#).

## 3) Etape 3 : Téléchargez le répertoire de travail

- ❖ Téléchargez le répertoire [udemy streamlit](#) de github et importez-le dans votre IDE.

### 4) Etape 4 : Création et activation d'un environnement virtuel

- ❖ L'environnement virtuel va nous permettre d'isoler le projet afin d'exécuter l'application avec les bonnes versions des librairies python.

Suivez ces étapes successives :

1. Placez vous dans le répertoire: `cd udemy_streamlit`
2. `pip install virtualenv` (si vous n'avez pas déjà virtualenv d'installer)
3. `virtualenv env` (création de l'environnement virtuel : env)
4. `source env/bin/activate` (activation de l'environnement virtuel)
5. `pip install -r requirements.txt` (installation des dépendances python)

### 5) Etape 5 : Créer un compte [github](#) + un compte [streamlit](#)

- ❖ Cela sera utile pour déployer notre application à la toute fin.

## 2/ Les bases de Streamlit.io



## ❖ Exécuter une application streamlit :

→ Doc [Streamlit](#)

- `streamlit run path_to_file/python_script.py`

## ❖ Flow de développement :

- Chaque fois que vous modifiez votre fichier python, il suffit d'enregistrer votre fichier pour que l'application se mette à jour.
- Chaque fois qu'un paramètre est modifié sur l'écran d'affichage de l'application, streamlit **ré exécute l'ensemble du script python de haut en bas.**

## ❖ Concept majeur :

- Programmation orienté objet : Pour modifier l'affichage de l'application il faut faire appel à l'objet streamlit (`st` dans notre cas) ainsi que la méthode utilisée.

### Exemple :

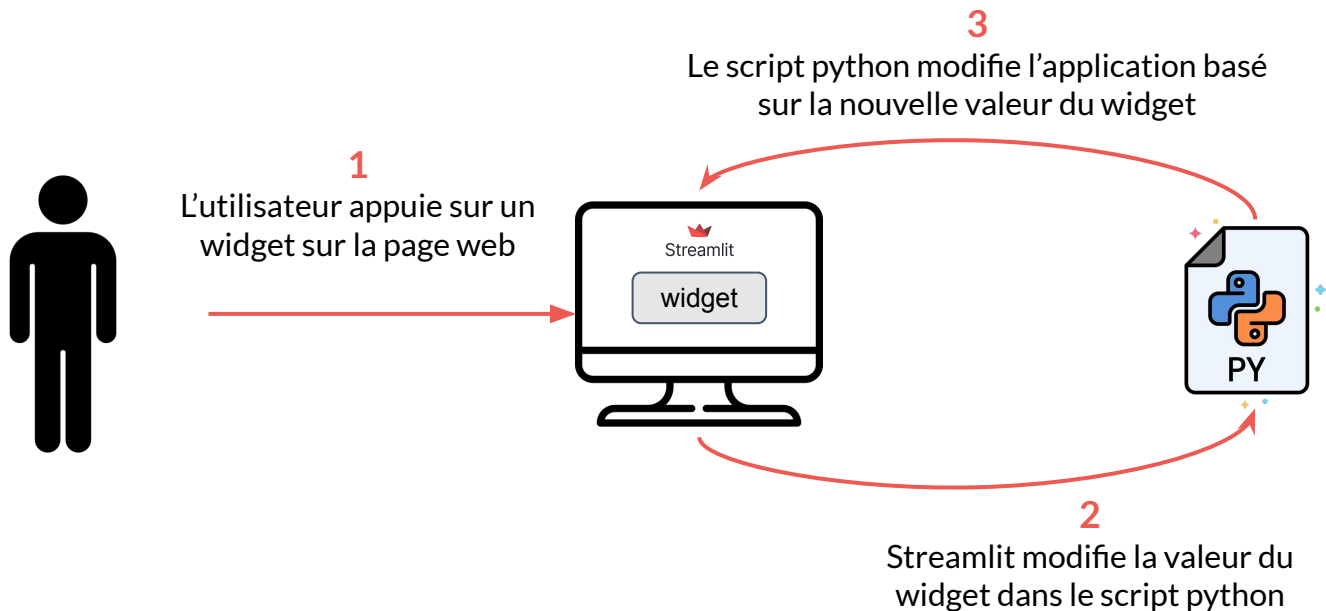
- `st.write("some text")` → affiche le texte "some text".
- `st.dataframe(df)` → affiche de la DataFrame df.

## 3/ Interaction utilisateur



L'interaction **Humain - Machine** sur [Streamlit](#) se fait à partir des **widgets**

(Boutton, Checkbox, SelectBox, Multiselect...)



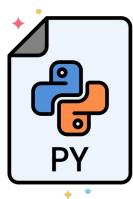
## 4/ Visualisation de graphs



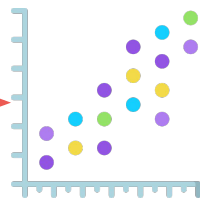


Pour créer un graphique sur une page web avec [Streamlit](#), il suffit de le créer en Python avec une librairie standard et d'appeler la fonction Streamlit adéquate.

*(Plotly, Matplotlib, Altair...)*



**Plotly**  
(par exemple)



`plotly.scatter()`

`st.plotly_chart()`



## 5/ Features avancées





## Form



Problème : Une application streamlit se ré exécute en entier dès qu'un utilisateur change la valeur d'un paramètre (widget)



Solution : **Streamlit Form**

→ `st.form` : Permet de créer un formulaire avec plusieurs widgets à l'intérieur. L'utilisateur peut changer la valeur des widgets sans que l'app se ré exécute.

→ `st.form_submit_button` : Bouton qui valide les valeur des widgets et ré exécute l'app qu'une seule fois !

Exemple



## Session



Problème : A chaque ré-exécution, l'ensemble des variables sont réinitialisées. Impossible de partager les valeurs des variables entre 2 runs.



Solution : **Streamlit Session State**

→ `st.session_state.ma_variable` : Permet de sauvegarder la variable `ma_variable` entre plusieurs ré-exécution de la session d'un même utilisateur.

Exemple dans les exercices



## Cache



Problème : Chaque fonction du code est exécutée à chaque nouveau run alors que ce n'est pas nécessaire si ses paramètres n'ont pas changé.



Solution : **Streamlit Cache**

→ `@st.cache` : Décorateur à placer au dessus d'un fonction pour éviter de la ré exécuter à chaque run. Très utile pour les fonctions de chargement de vos données (qui sont les mêmes d'un run à l'autre)

La fonction se ré exécute en partie seulement si les paramètres en entrée ont changé.

Exemple dans les exercices



## 6/ Déployer son application



## Comment rendre son application accessible sur internet partout dans le monde ?

### 1) Etape 1 : Créer un répertoire sur Github avec l'ensemble de votre projet

- ❖ Il suffit de vous connecter à votre compte github et d'uploader vos fichiers dans un nouveau répertoire.

### 2) Etape 2 : Connectez vous à [Streamlit](#) et donnez l'accès à votre compte github

- ❖ Streamlit va s'occuper de déployer l'application qui se situe dans votre répertoire Github.

### 3) Etape 3 : Déployez votre application !

- ❖ Cliquez sur : "New App" en haut à droite.
- ❖ Remplissez les paramètres du formulaire (Nom du répertoire Github / la branche / Nom du fichier).
- ❖ Cliquez sur "Advanced Settings" : Choisissez votre version de Python.
- ❖ Cliquez sur "Deploy" et attendez quelques minutes que l'application se déploie.
- ❖ 🎉 Si tout se passe bien, votre application est créée ! 🎉

# Conclusion



## Félicitations !

**Vous savez maintenant développer et rendre accessible une application web !**

- Si le cours vous a plu, vous pouvez le noter et me laisser un commentaire, cela me permettra de m'améliorer pour les prochains tutoriels. 👍



pl.danieau@gmail.com



Pierre-Louis Danieau,  
Ingénieur Data Scientist

PS : J'explique dans cet article Medium comment j'ai développé avec Streamlit une application de réservation de trains utilisés par plus de 10 000 personnes !