

Guide d'installation et configuration de facebook prophet dans Python et R sur un environnement Linux (Ubuntu).

Avant de commencer

D'abord on met à jour la base de données des paquets et on les met à jour si nécessaire.

```
sudo apt update
```

```
sudo apt upgrade (facultative)
```

Au besoin vous pouvez cloner le repo *git* contenant ce guide et un exemple de l'utilisation de *prophet*.

```
git clone https://github.com/briones-csfoy/fbprophet-installation.git
```

La guide d'installation officiel se trouve dans la suivante page internet :

<https://facebook.github.io/prophet/docs/installation.html>

Python

Nous devons avoir installé *Python 3.x* et *pip*, *Jupyter* sera nécessaire pour exécuter l'exercice proposé. Il est recommandé d'installer *prophet* dans un environnement virtuel donc la librairie *venv* est conseillé. La suivante instruction peut être utilisée à ce propos :

```
sudo apt install python3 python3-pip python3-venv jupyter
```

La librairie *ipykernel* sera utilisé pour ajouter le noyau à *Jupyter*. Cela peut être installé dans l'environnement virtuel, mais vous pouvez le faire maintenant avec la suivante instruction et la rendre disponible pour n'importe quel environnement :

```
pip3 install ipykernel
```

Le suivant étape est créer et activer l'environnement virtuel, je l'ai nommé *prophet*, mais vous pouvez choisir le nom que vous conviennent :

```
python3 -m venv ~/prophet
```

```
source ~/prophet/bin/activate
```

Pour éviter des erreurs pendant l'installation ou l'utilisation de *prophet* on doit installer les librairies *wheel* et *plotly*, et pour notre exemple *nbformat* et *pandas* sont requises.

```
pip install wheel
```

```
pip install plotly
```

```
pip install nbformat
```

```
pip install pandas
```

Si vous avez cloné le repo git proposé de-sous, vous pouvez utiliser la suivante instruction au lieu des quatre dernières :

```
pip install -r ~/fbprophet-installation/requirements.txt
```

La plus importante dépendance de *prophet* est *pystan*, donc il faut l'installer :

```
pip install pystan
```

Finalement vous installez *prophet* dans l'environnement virtuel avec la suivante instruction :

```
pip install fbprophet
```

Pour valider l'installation vous pouvez importer *fbprophet* dans l'invite de commandes de *python*, si aucun message est reçu, l'installation est un succès.

```
python
```

```
>>> from fbprophet import Prophet
```

```
>>> exit()
```

Si une erreur se produit on peut se essayer en désinstallant et installant à nouveau *prophet* mais peut être nécessaire de vérifier que la librairie *convertdate* soit dans notre environnement.

```
pip install convertdate
```

```
pip uninstall fbprophet
```

```
pip install fbprophet
```

Exemple sur Python

Si vous n'avez pas encore installé ipykernel, vous pouvez le faire dans l'environnement virtuel :

```
pip install ipykernel
```

Maintenant vous devez ajouter le noyau (environnement virtuel) à *jupyter notebook* (assurez-vous que l'environnement où vous avez installé *prophet* est active):

```
python -m ipykernel install --user --name prophet --display-name "Prophet"
```

Dans cet exemple j'ai choisi le nom « prophet » pour le noyau, mais vous pouvez le changer à votre goût.

Si vous avez décidé de ne pas cloner le repo *git*, vous pouvez télécharger le fichier csv et visiter la page du quick-start de *facebook prophet* pour exécuter le code d'exemple dans *jupyter notebook* :

```
wget
```

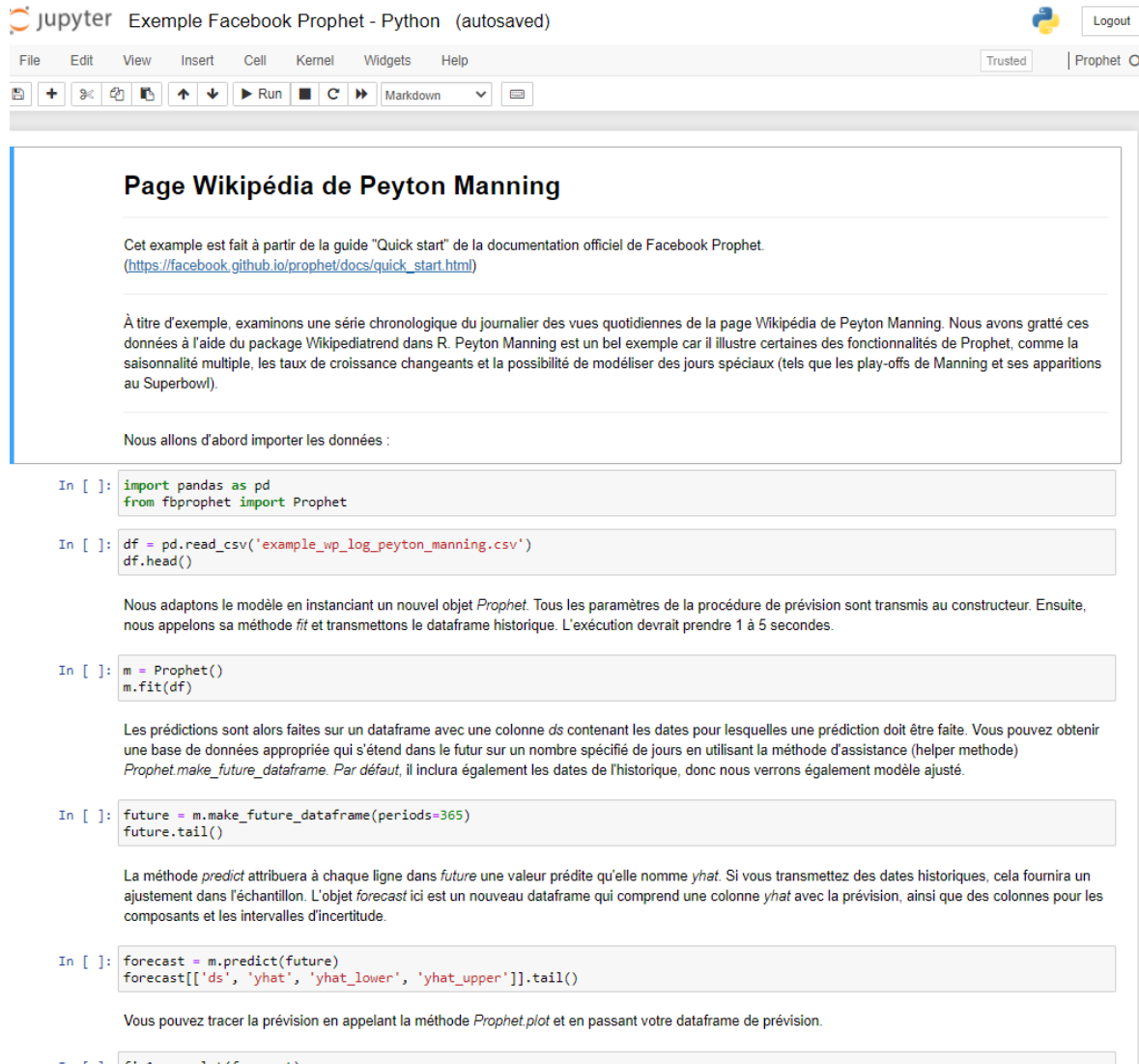
```
https://raw.githubusercontent.com/facebook/prophet/master/examples/example_wp_log_peyton_manning.csv
```

https://facebook.github.io/prophet/docs/quick_start.html

Si vous avez cloné le repo git vous pouvez lancer jupyter notebook, puis aller au repertoire `~/fbprophet-installation/exemples/` et exécuter le fichier « Exemple Facebook Prophet - Python.ipynb ».

Jupyter notebook

Normalement vous pouvez accéder à *jupyter notebook* avec un navigateur web à partir de l'adress `htt://localhost:8888`



Page Wikipédia de Peyton Manning

Cet exemple est fait à partir de la guide "Quick start" de la documentation officiel de Facebook Prophet.
(https://facebook.github.io/prophet/docs/quick_start.html)

À titre d'exemple, examinons une série chronologique du journalier des vues quotidiennes de la page Wikipédia de Peyton Manning. Nous avons gratté ces données à l'aide du package Wikipediatrend dans R. Peyton Manning est un bel exemple car il illustre certaines des fonctionnalités de Prophet, comme la saisonnalité multiple, les taux de croissance changeants et la possibilité de modéliser des jours spéciaux (tels que les play-offs de Manning et ses apparitions au Superbowl).

Nous allons d'abord importer les données :

```
In [ ]: import pandas as pd
from fbprophet import Prophet
```

```
In [ ]: df = pd.read_csv('example_wp_log_peyton_manning.csv')
df.head()
```

Nous adaptons le modèle en instanciant un nouvel objet *Prophet*. Tous les paramètres de la procédure de prévision sont transmis au constructeur. Ensuite, nous appelons sa méthode *fit* et transmettons le dataframe historique. L'exécution devrait prendre 1 à 5 secondes.

```
In [ ]: m = Prophet()
m.fit(df)
```

Les prédictions sont alors faites sur un dataframe avec une colonne *ds* contenant les dates pour lesquelles une prédiction doit être faite. Vous pouvez obtenir une base de données appropriée qui s'étend dans le futur sur un nombre spécifié de jours en utilisant la méthode d'assistance (helper methode) *Prophet.make_future_dataframe*. Par défaut, il inclura également les dates de l'historique, donc nous verrons également modèle ajusté.

```
In [ ]: future = m.make_future_dataframe(periods=365)
future.tail()
```

La méthode *predict* attribuera à chaque ligne dans *future* une valeur prédite qu'elle nomme *yhat*. Si vous transmettez des dates historiques, cela fournira un ajustement dans l'échantillon. L'objet *forecast* ici est un nouveau dataframe qui comprend une colonne *yhat* avec la prévision, ainsi que des colonnes pour les composants et les intervalles d'incertitude.

```
In [ ]: forecast = m.predict(future)
forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']].tail()
```

Vous pouvez tracer la prévision en appelant la méthode *Prophet.plot* et en passant votre dataframe de prévision.

```
In [ ]: fig1 = m.plot(forecast)
```

R

Vous pouvez installer *R* en suivant la documentation officielle (<https://utstat.toronto.edu/cran/bin/linux/ubuntu/README>)

```
sudo apt update -qq
```

```
sudo apt install --no-install-recommends software-properties-common dirmngr
```

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
E298A3A825C0D65DFD57CBB651716619E084DAB9
```

```
sudo add-apt-repository "deb https://cloud.r-project.org/bin/linux/ubuntu  
$(lsb_release -sc)-cran40/"
```

```
sudo apt install --no-install-recommends r-base
```

Assurez-vous d'avoir installé R pour développeurs car il sera nécessaire pour compiler le paquet *prophet* et ses dépendances, au besoin vous pouvez l'installer à l'aide de la suivante instruction :

```
sudo apt install r-base-dev
```

Deux librairies sont aussi essentielles : *libcurl* et *libv8* donc il faut les installer.

```
sudo apt install libcurl4-openssl-dev
```

```
sudo apt install libv8-dev
```

Dans l'invite de commandes R on installe *prophet*, une seule instruction est nécessaire pour le faire :

```
R
```

```
> install.packages('prophet')
```

Le processus prend plusieurs minutes. Vous pouvez valider que l'installation est correcte en important la librairie *prophet* :

```
> library(prophet)
```

Si aucun message d'erreur s'affiche l'installation est un succès.

Après vous pouvez ajouter le noyau R à *jupyter* et exécuter l'exemple proposé. Pour cela il faut installer la librairie *IRkernel* puis rendre disponible le noyau à l'aide de cette librairie :

```
> install.packages('IRkernel')
```

```
> IRkernel::installspec()
```

Évidemment vous pouvez utiliser *RStudio* ou bien autre outil pour lancer votre projet de prédiction avec *prophet*, mais l'exemple inclus dans le repo *git* s'exécute sur *jupyter notebook*.

Si vous avez décidé de ne pas cloner le repo *git*, vous pouvez télécharger le fichier csv et visiter la page du quick-start de facebook *prophet* pour exécuter le code d'exemple dans *jupyter notebook* :

```
wget
```

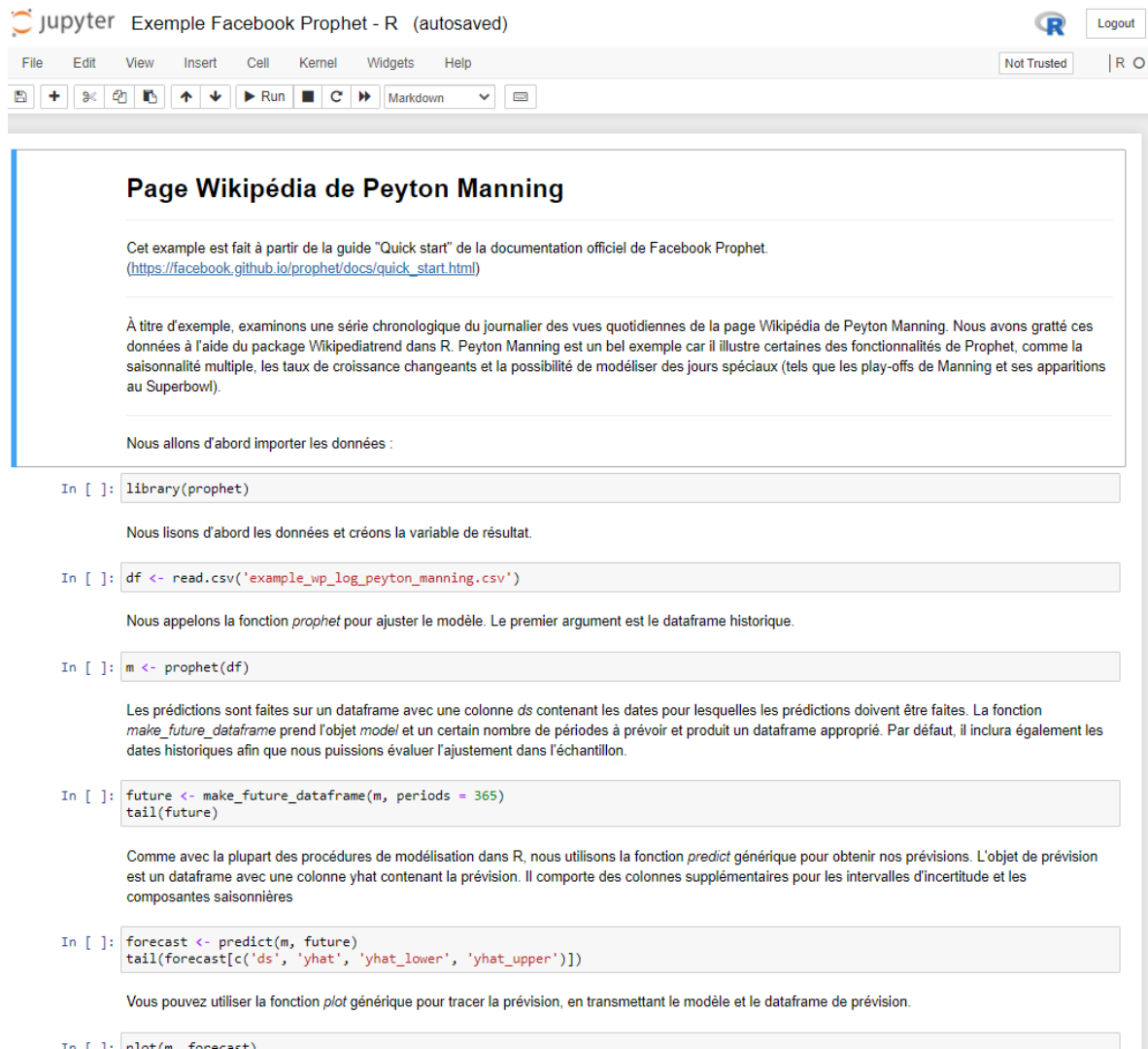
```
https://raw.githubusercontent.com/facebook/prophet/master/examples/example_wp_lo  
g_peyton_manning.csv
```

https://facebook.github.io/prophet/docs/quick_start.html

Si vous avez cloné le repo *git* vous pouvez lancer *jupyter notebook*, puis aller au repertoire `~/fbprophet-installation/exemples/` et exécuter le fichier « Exemple Facebook Prophet - R.ipynb ».

Jupyter notebook

Normalement vous pouvez accéder à *jupyter notebook* avec un navigateur web à partir de l'adress `htt://localhost:8888`



Page Wikipédia de Peyton Manning

Cet exemple est fait à partir de la guide "Quick start" de la documentation officiel de Facebook Prophet.
(https://facebook.github.io/prophet/docs/quick_start.html)

À titre d'exemple, examinons une série chronologique du journalier des vues quotidiennes de la page Wikipédia de Peyton Manning. Nous avons gratté ces données à l'aide du package Wikipediatrend dans R. Peyton Manning est un bel exemple car il illustre certaines des fonctionnalités de Prophet, comme la saisonnalité multiple, les taux de croissance changeants et la possibilité de modéliser des jours spéciaux (tels que les play-offs de Manning et ses apparitions au Superbowl).

Nous allons d'abord importer les données :

```
In [ ]: library(prophet)
```

Nous lisons d'abord les données et créons la variable de résultat.

```
In [ ]: df <- read.csv('example_wp_log_peyton_manning.csv')
```

Nous appelons la fonction *prophet* pour ajuster le modèle. Le premier argument est le dataframe historique.

```
In [ ]: m <- prophet(df)
```

Les prédictions sont faites sur un dataframe avec une colonne *ds* contenant les dates pour lesquelles les prédictions doivent être faites. La fonction *make_future_dataframe* prend l'objet *model* et un certain nombre de périodes à prévoir et produit un dataframe approprié. Par défaut, il inclura également les dates historiques afin que nous puissions évaluer l'ajustement dans l'échantillon.

```
In [ ]: future <- make_future_dataframe(m, periods = 365)
tail(future)
```

Comme avec la plupart des procédures de modélisation dans R, nous utilisons la fonction *predict* générique pour obtenir nos prévisions. L'objet de prévision est un dataframe avec une colonne *yhat* contenant la prévision. Il comporte des colonnes supplémentaires pour les intervalles d'incertitude et les composantes saisonnières

```
In [ ]: forecast <- predict(m, future)
tail(forecast[c('ds', 'yhat', 'yhat_lower', 'yhat_upper')])
```

Vous pouvez utiliser la fonction *plot* générique pour tracer la prévision, en transmettant le modèle et le dataframe de prévision.

```
In [ ]: plot(m, forecast)
```