

# Online Shoppers INTENTIONS

Peut-on prévoir vos achats ?



# Sommaire

01

## Pré-processing

- Description du dataset
- Transformation des données

02

## Analyse des variables

- Analyse des corrélations
- Analyses bivariées
- Analyses multivariées

03

## Modèles de prédictions

- Machine Learning
- Deep Learning

04

## Opérationnel

- Transformation des modèles en API Flask
- Conseils pour votre site d'e-commerce



*“The world is now awash in data  
and we can see consumers in a lot  
clearer way.”*

Max **Levchin**

01

# Pré- processing

**Torture the data,  
and it will confess to anything.**

- Ronald Coase  
British Economist and Author



# Attributs



Revenue

## Durée par type (3)



- Administrative Duration
- Informational Duration
- Product Related Duration



## Type de page (3)

- Administrative
- Informational
- Product related



## Temporalité (3)

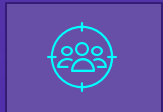
- Month
- Special day
- Week-end

# Attributs



## Rates (2)

- Bounce Rates
- Exit Rates

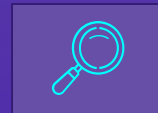


## Page Values



## Session informatique (3)

- Traffic Type
- Browser
- OperatingSystems



## Autre (2)

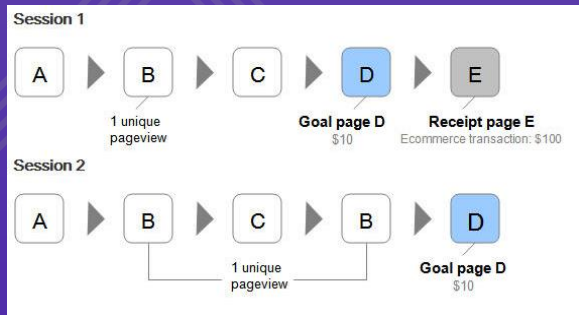
- Visitor Type
- Region

# Page Value ou valeur d'une page



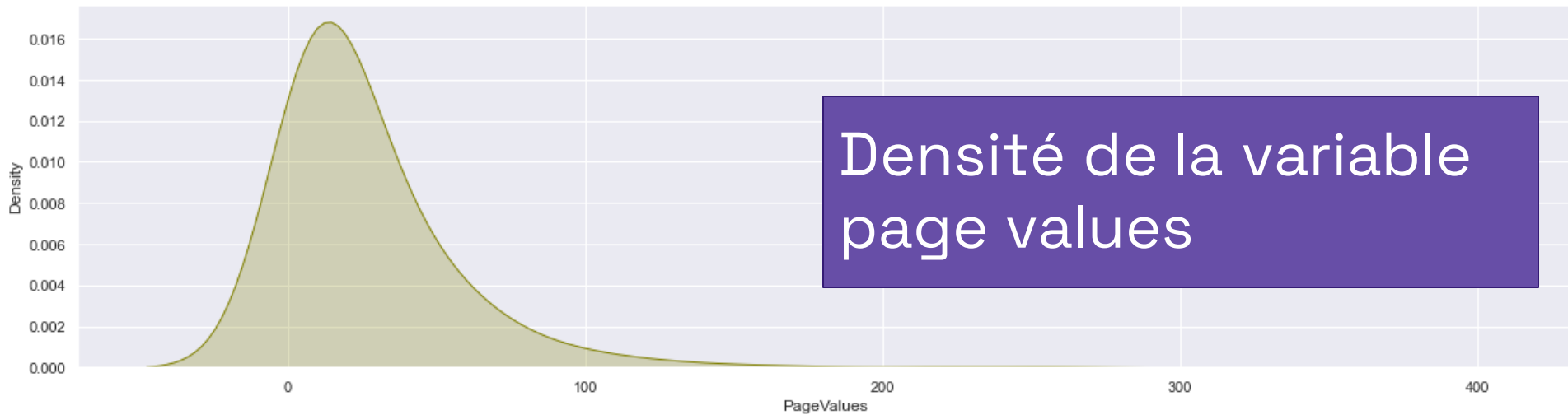
Chiffre d'Affaire généré par le e-commerce

Nombre vues uniques de la page

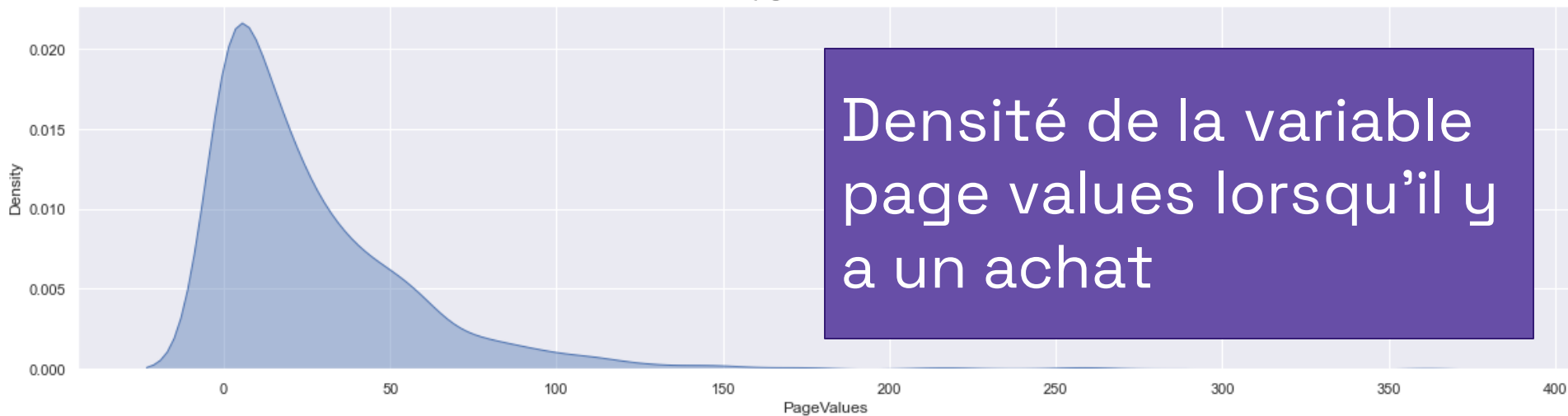


$$\text{Page Value de B} = (2 \times 10 + 100) / 2 = 60$$

Pages values - Densité

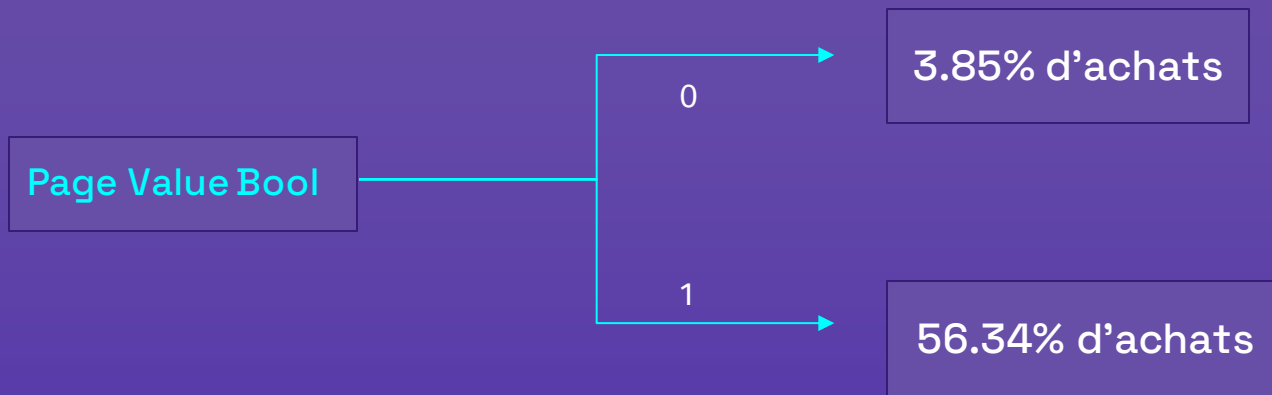


Densité de pageValue sur sessions d'achat





# Création de PageValueBool



**15% de probabilité d'achat pour l'ensemble des données**

On obtient une corrélation de 0.6 entre PageValueBool et Revenu  
Nous n'avons que 0.49 avec PageValue.

On en déduit que cette nouvelle colonne aura un meilleur impact dans nos modèles

# Encodage et classification

## Variables multiclass

- Type de visiteur
- Mois

## Variables binaires

- Revenue
- Week-End

## Variables > 0

- Product Related
- Administrative
- Informational

18 variables



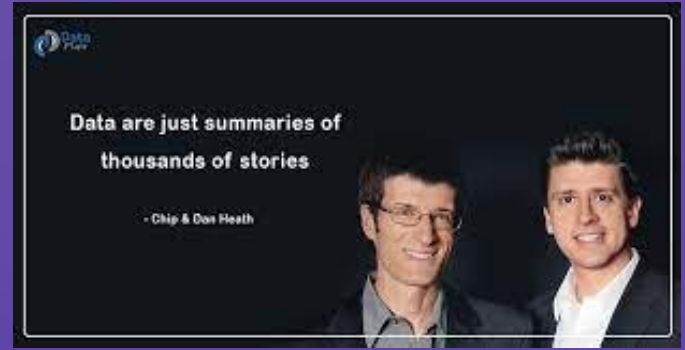
35 variables

## Standard scaler

$$z = \frac{x - \mu}{\sigma}$$

# 02

## Analyse des variables



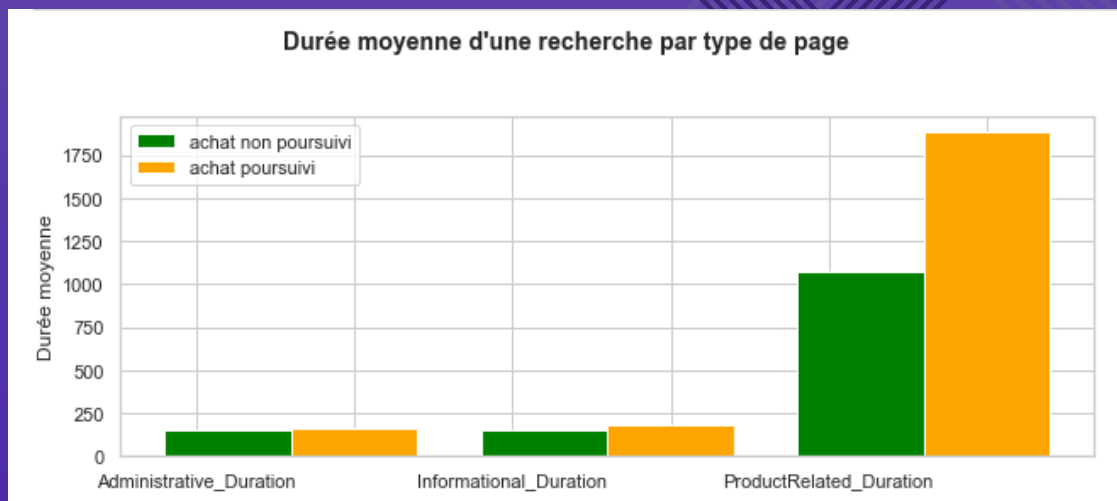
# Analyses bivariées et multivariées

*Analyse de différentes variables en fonction de notre target 'Revenue'*

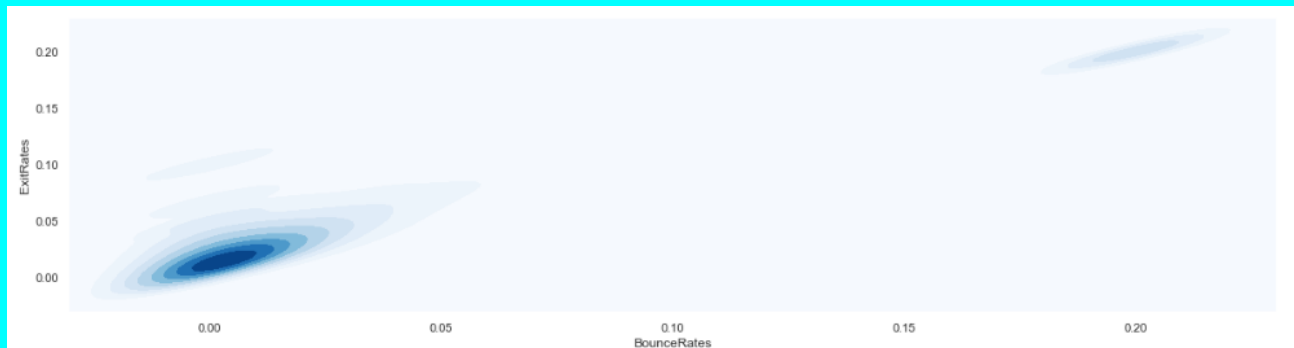
*La durée passée sur une page influe-t-elle sur l'achat ?*

Moyenne des durées selon l'achat

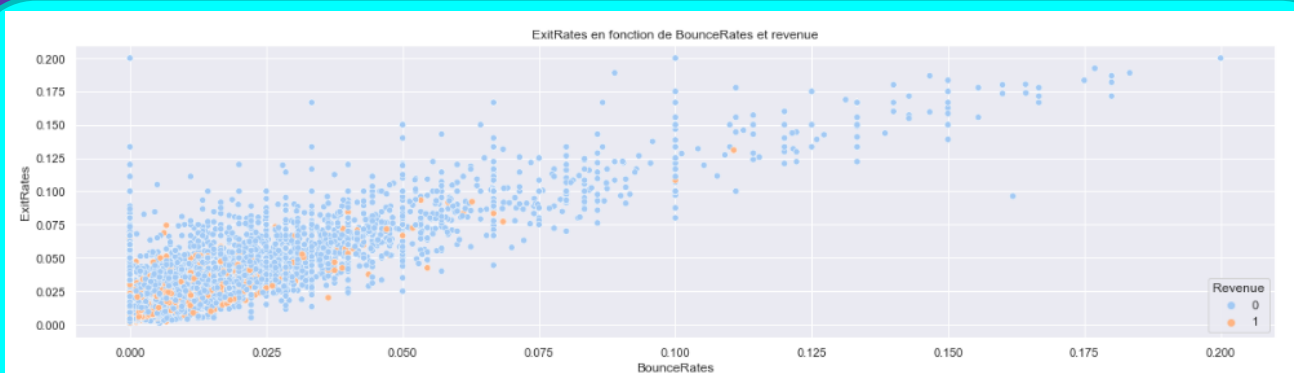
	0	1
Administrative_Duration	148.707322	163.539476
Informational_Duration	156.155632	179.319090
ProductRelated_Duration	1073.283248	1882.128257



## Il y a-t-il un lien entre les variables exit et bounce rates?

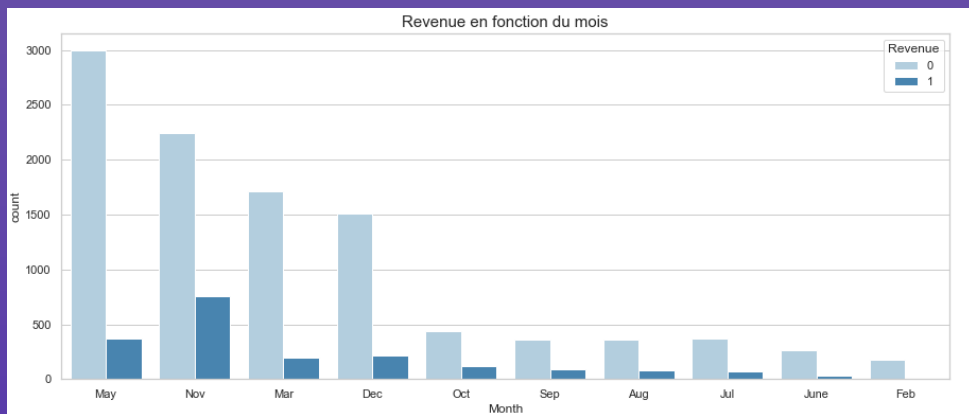


Coefficient de Pearson  
entre BounceRate et  
ExitRate = 0.91



Cela indique donc que  
lorsqu'une page web a  
un taux de rebond  
faible, elle a également  
un taux de sortie  
faible. Il y a de grandes  
chances que cette  
page mène à un achat

# Le mois de la recherche influe-t-il sur l'achat ?



On observe que c'est au mois de novembre qu'une session a le plus de chance de finir sur un achat.

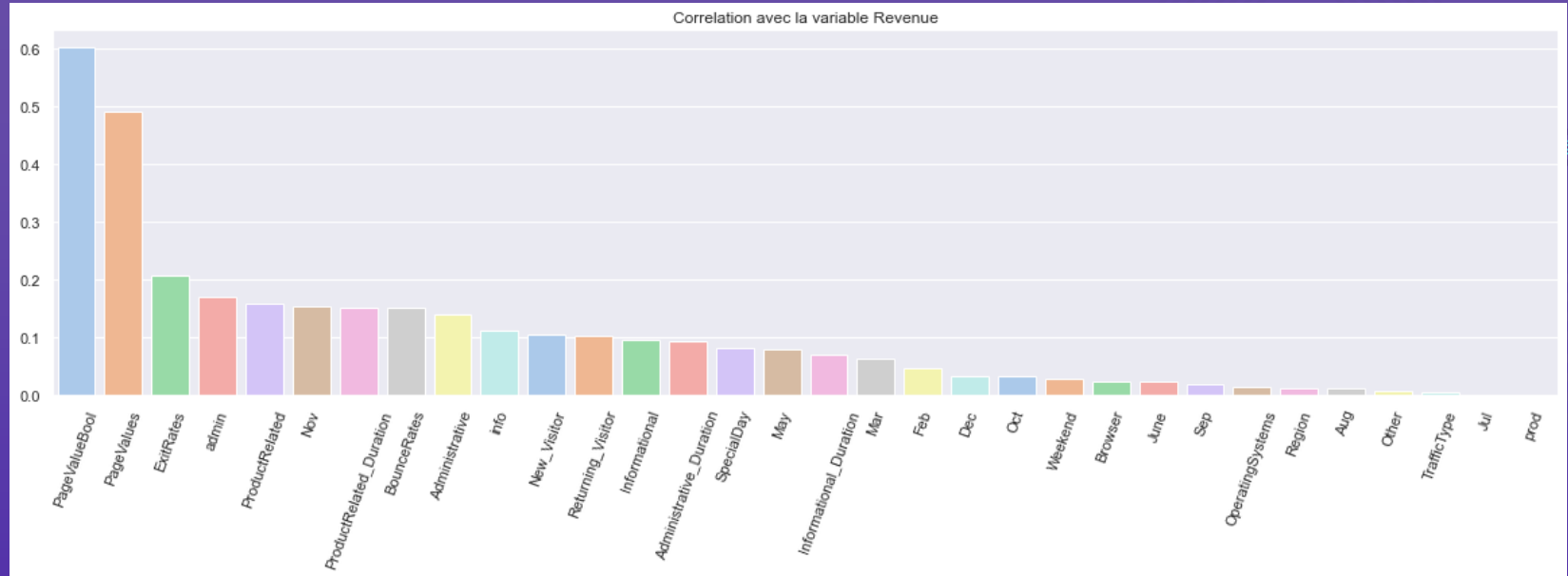
En effet, au mois de novembre, il y a 25.3% de chance que la session aboutisse à un achat.

Y-a-t-il un lien avec la variable special day ?  
Non.

En effet, en regroupant par jours spéciaux, on se rend compte avec notre variable ratio (%) que 1831 achats se font avec des SpecialDay de 0, soit 95.96% des achats totaux.

	SpecialDay	Revenue_x	prod_x	Revenue_y	prod_y	ratio
4	0.8	1	11	0	314	3.384615
2	0.4	1	13	0	230	5.349794
5	1.0	1	10	0	144	6.493506
1	0.2	1	14	0	164	7.865169
3	0.6	1	29	0	322	8.262108
0	0.0	1	1831	0	9248	16.526762

# Quelles sont les variables les plus corrélées à Revenue?



- 1 - PageValueBool : 0.6
- 2 - Page value : 0.49
- 3 - ExitRates : -0.21
- 4 - Admin : 0.17

- 5 - Product Related : 0.16
- 6 - Nov : 0.15
- 7 - Product Related Duration : 0.15
- 8 - BounceRates : -0.15



# 03

## Modèles de prédiction

“Where there is data smoke,  
there is business fire.”

Thomas Redman  
aka “the Data Doc”



# Division du dataset

Page Value Bool

Page Values

Exit Rates

Product Related

Nov

Product related Duration

Bounce Rates

Admin

80% ~ 9900

X\_Train

20% ~ 2500

X\_Test



Deep Learning

[False  
True  
False  
False  
False]

OneHotEncoder()

Y

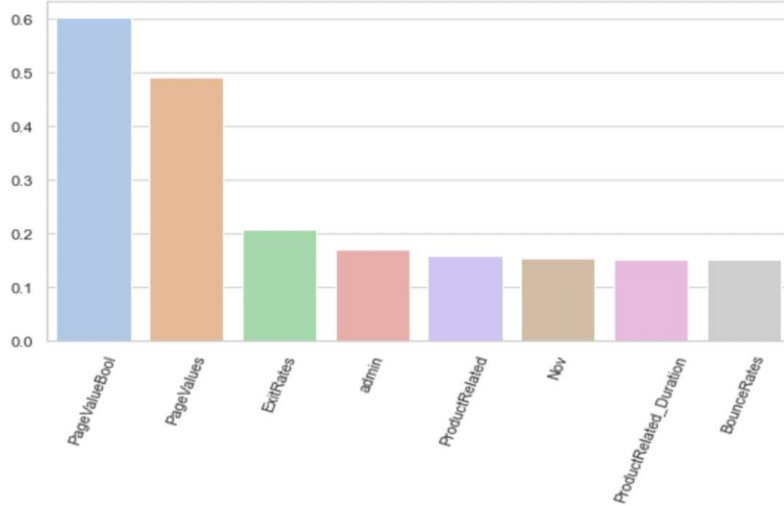
[[0,1]  
[1,0]  
[0,1]  
[0,1]  
[0,1]]

# Random Forest

	Test	Choix
N estimators	<ul style="list-style-type: none"><li>• 100</li><li>• 200</li><li>• 500</li></ul>	100
Max features	<ul style="list-style-type: none"><li>• Auto</li><li>• Sqrt</li></ul>	Sqrt
Max Depth	<ul style="list-style-type: none"><li>• 4</li><li>• 6</li><li>• 8</li></ul>	8
Criterion	<ul style="list-style-type: none"><li>• Gini</li><li>• Entropy</li></ul>	entropy

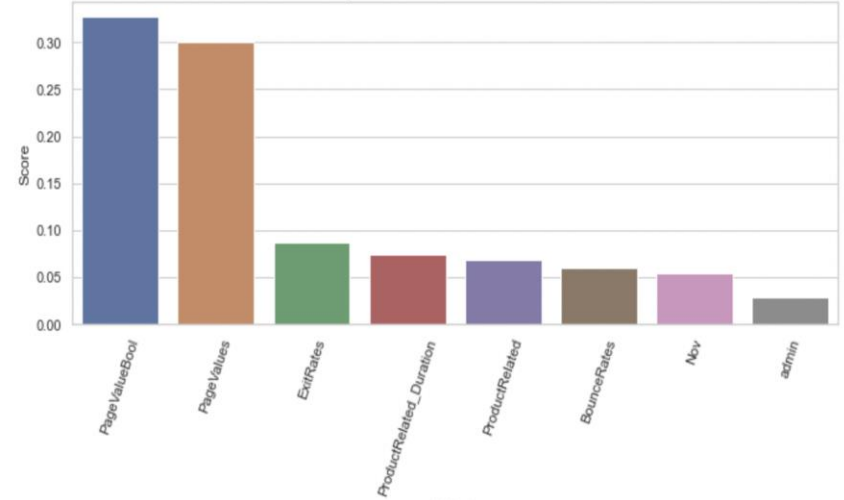
# Random Forest

Correlation avec la variable Revenue



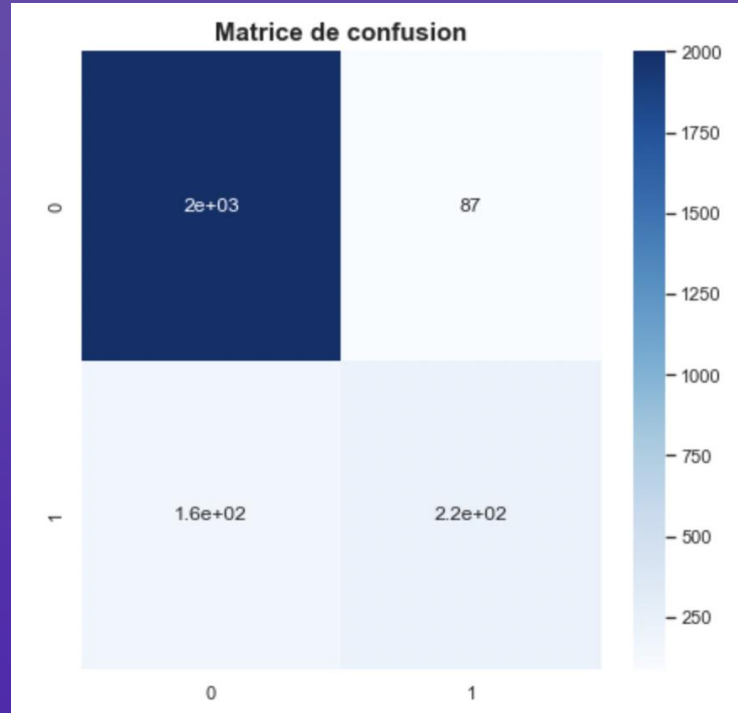
Correlation des  
variables à "Revenue"

Importance des attributs



Importance des  
variables du modèle

# Random Forest



# Random Forest

## Estimateurs

Jaccard Score - REVENUE = 0 : 0.8922048997772829  
Jaccard Score - REVENUE = 1: 0.4773218142548596  
F1 SCORE: 0.8977727784759437

	precision	recall	f1-score	support
0	0.93	0.96	0.94	2090
1	0.72	0.59	0.65	376
accuracy			0.90	2466
macro avg	0.82	0.77	0.79	2466
weighted avg	0.90	0.90	0.90	2466

# Autres modèles

	MSE	MAE	R2	Jacc_score0	Jacc_score1	F1
Algorithm						
Random Forest	0.09	0.09	0.26	0.90	0.48	0.90006
SVC	0.10	0.10	0.25	0.89	0.48	0.89885
KNN_Class	0.10	0.10	0.24	0.89	0.46	0.89656
Logistic_regression	0.10	0.10	0.24	0.89	0.46	0.89635
XGBoost	0.10	0.10	0.24	0.89	0.46	0.89571
Decision_tree	0.10	0.10	0.19	0.89	0.43	0.88808
GaussianNB_Class	0.12	0.12	0.06	0.87	0.41	0.87658

# Deep Learning

- Utilisation de tensorflow et Keras
- 2 modèles possible, ici en modèle séquentiel





# Comment créer un modèle simple ?

```
modele.add(Dense(nb_neurones, activation=ma_fonction))
```

1ère couche : les entrées

- Les caractéristiques de la forme d'entrée

Couches intermédiaires

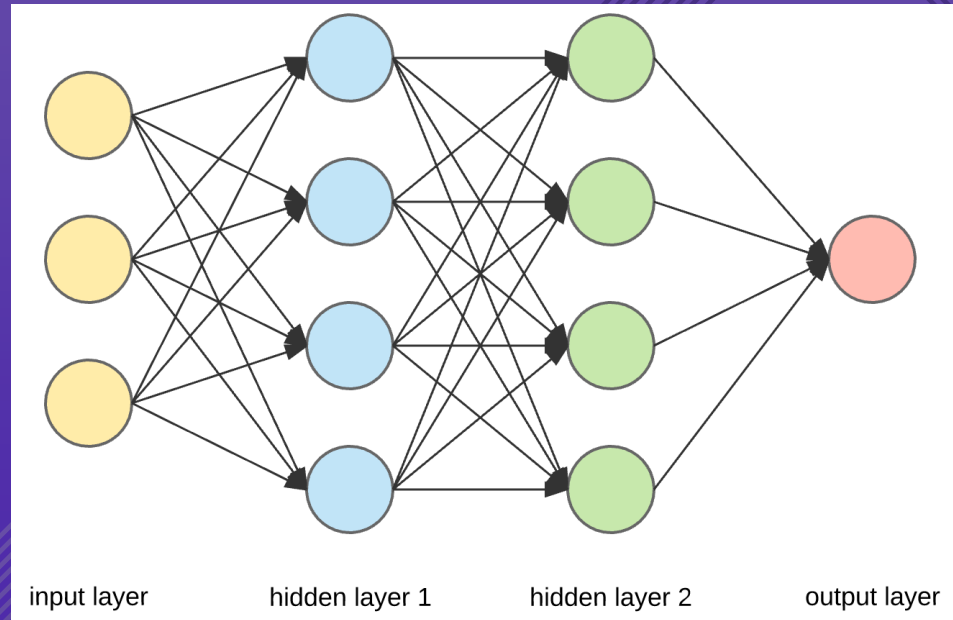
- Connectées aux autres couches

Dernière couche = couche de sortie

- Les caractéristiques de la forme de sortie

# Exemple simple

- Première couche 3 entrées
  - 2e couche : 4 neurones
  - 3e couche : 4 neurones
  - 4e couche : 1 sortie
- 
- Tous les neurones d'entrées sont connectés à tous les neurones de sorties



# Notre modèle

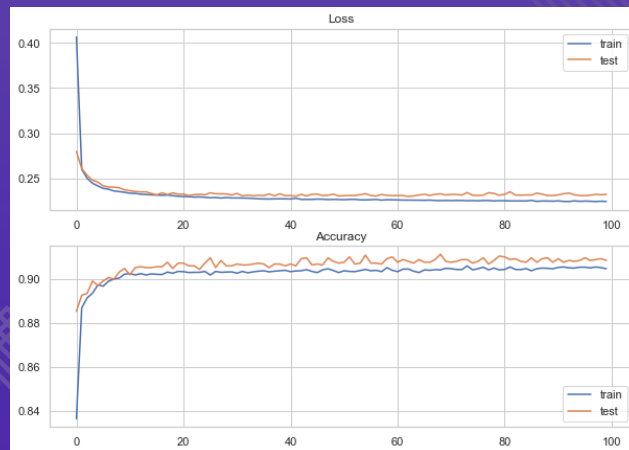
- 3 couches de calculs
- 1ere couche : 8 entrées
- Fonction 'Relu' pour les couches intermédiaires
- Couche de sortie en sigmoide
- Compilateur : loss binary crossentropy
- Metrics : accuracy

F1 score = 0.90

```
deepmodel = keras.Sequential(  
    [  
        layers.Dense(units=8, input_dim=8, activation="relu", name="layer1"),  
        layers.Dense(units=32, activation="relu", name="layer2"),  
        layers.Dense(units=2, activation="sigmoid", name="layer3"),  
    ]  
)  
  
deepmodel.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Model: "sequential"

Layer (type)	Output Shape	Param #
layer1 (Dense)	(None, 8)	72
layer2 (Dense)	(None, 32)	288
layer3 (Dense)	(None, 2)	66
Total params: 426		
Trainable params: 426		
Non-trainable params: 0		



# Quel est le meilleur modèle?

	MSE	MAE	R2	Jacc_score0	Jacc_score1	F1
DeepLearning	NaN	NaN	NaN	NaN	NaN	0.90361
Random Forest	0.09	0.09	0.26	0.90	0.48	0.90006
SVC	0.10	0.10	0.25	0.89	0.48	0.89885
KNN_Class	0.10	0.10	0.24	0.89	0.46	0.89656
Logistic_regression	0.10	0.10	0.24	0.89	0.46	0.89635
XGBoost	0.10	0.10	0.24	0.89	0.46	0.89571
Decision_tree	0.10	0.10	0.19	0.89	0.43	0.88808
GaussianNB_Class	0.12	0.12	0.06	0.87	0.41	0.87658

# Et avec tout le dataset, ça donne quoi ?

```
deepmodel2 = keras.Sequential([
    layers.Dense(units=30, input_dim=30, activation="relu", name="layer1"),
    layers.Dense(units=64, activation="relu", name="layer2"),
    layers.Dense(units=2, activation="sigmoid", name="layer3"),
])

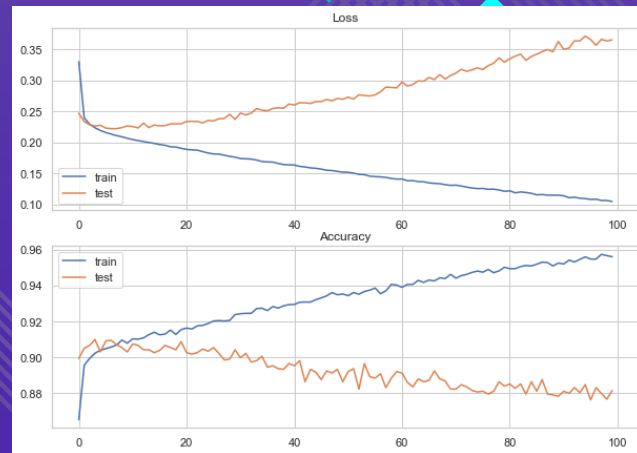
deepmodel2.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
deepmodel2.summary()

save2 = deepmodel2.fit(X_train, y_train, epochs=100, validation_data=(X_test, y_test))
```

Model: "sequential\_2"

Layer (type)	Output Shape	Param #
layer1 (Dense)	(None, 30)	930
layer2 (Dense)	(None, 64)	1984
layer3 (Dense)	(None, 2)	130
Total params: 3,044		
Trainable params: 3,044		
Non-trainable params: 0		

- Haute accuracy sur le train (+0.95)
  - Faible sur le test (0.87)
- => Sur-apprentissage ou "*overfitting*" !



04

API



<http://localhost:9000/?mod=KNN&pvBool=1&pv=1&exiRa=1&prodRel=1&nov=1&prodRelDur=1&bounRa=1&admi=1>

Connexion locale

BEST  
ALL  
KNN  
GNB  
SVC  
DT  
LR  
XGB  
RF  
DEEP

Les 8 variables, séparées ou dans /s  
La valeur par défaut de chaque variable est **sa moyenne** dans le train

**Retourne 0 ou 1** (Pour être implémenté dans un autre code)

Sauf pour ALL qui retourne un string :  
[nom du modèle] => valeur, ...



## NOS CONSEILS

- 
- Communiquer au mois de **novembre**
  - Plus de pages **produits**
  - Peu de pages **administratives**
  - Conservation des clients (**bounce & exit rates, duration**)





MERCI POUR VOTRE ECOUTE