Estimer le salaire d’un informaticien

Presenté par :
PLANTADE Adrien, BLAY Adrien
IUT informatique de Clermont Ferrand 2e année

Sommaire

I.Explication du sujet

- Données d’un sondage de plus de 51000 votes sur 143 questions.

- Peut-on estimer le salaire d’un informaticien avec ces données ?

II. Langage et bibliothèques utilisés

- Python

- Numpy
- Pandas
- Matplotlib
- Keras

·Pandas

In [38]:

```
DataYears3=DataYears
DataYears3.to_numpy()
dy=np.transpose(DataYears3)
dy
df3=pd.DataFrame(dy)
df3
```

Out [38]:

	Salary	YearsProgram
2	113750	20 or more years
14	100000	20 or more years
17	130000	20 or more years
22	100764	10 to 11 years
25	175000	20 or more years
...	...	...
51361	440.464	1 to 2 years

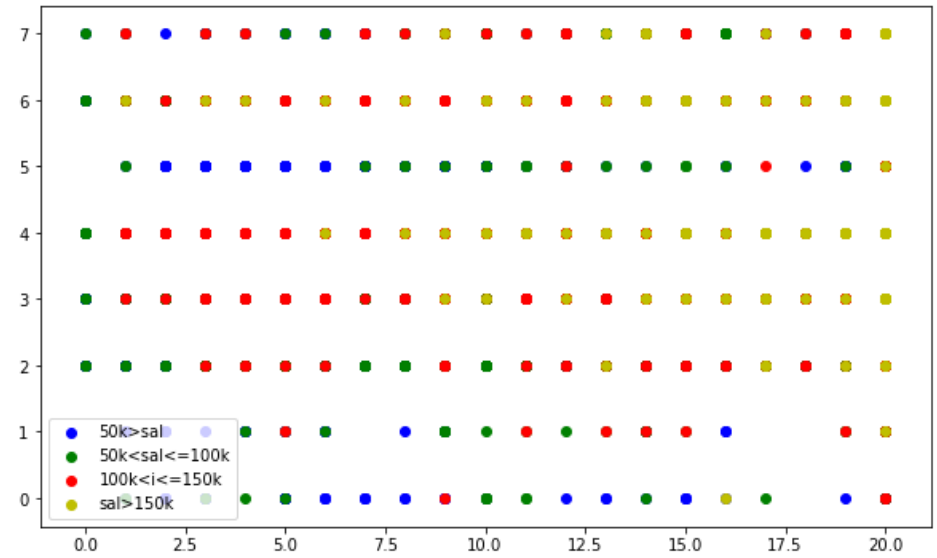
Numpy

```
print(dataFinalX)
#Les valeurs des classes
print(np.unique(dataFinal.get("catSal").values))
dataFinalY=dataFinal.get("catSal").values
```

```
[[20.  4.]
 [20.  5.]
 [20.  4.]
 ...
 [ 5.  6.]
 [ 6.  4.]
 [17.  4.]]
[0. 1. 2. 3.]
```

Matplotlib

```
#pour mieux comprendre notre data, on va visualiser des exemples
#visualisation des données
plt.figure(figsize=(10, 6))
plt.scatter(dataFinalX[dataFinalY == 3][:, 0], dataFinalX[dataFinalY == 3][:, 1], color='b', label='50k>sal')
plt.scatter(dataFinalX[dataFinalY == 2][:, 0], dataFinalX[dataFinalY == 2][:, 1], color='g', label='50k<sal<=100k')
plt.scatter(dataFinalX[dataFinalY == 1][:, 0], dataFinalX[dataFinalY == 1][:, 1], color='r', label='100k<i<=150k')
plt.scatter(dataFinalX[dataFinalY == 0][:, 0], dataFinalX[dataFinalY == 0][:, 1], color='y', label='sal>150k')
plt.legend();
```



Keras

```
from keras.models import Sequential
from keras.layers import Dense
#input (X) and output (y) variables
X = dataFinal.to_numpy()[:,0:2] #Les années d'expérience et la catégorie d'études
y = dataFinal.to_numpy()[:,3] #La catégorie binaire de salaire, >100k et <100k
# define the keras model
# 3 couches de neurones
model = Sequential()
model.add(Dense(30, input_dim=2, activation='relu')) #30 neurones Rectified Linear Unit (ReLU) : renvoie une valeur selon
#son poids
model.add(Dense(20, activation='relu')) #20 neurones Rectified Linear Unit (ReLU)
model.add(Dense(1, activation='sigmoid'))  #1 sortie, 0 ou 1
# compile the keras model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the keras model on the dataset
model.fit(X, y, epochs=100, batch_size=20,verbose=1)
# evaluate the keras model
_, accuracy = model.evaluate(X, y)
print('Accuracy: %.2f' % (accuracy*100))


predictions = model.predict_classes(X)
# Exemple des 50 premiers cas
for i in range(50):
    print('%s => %d (expected %d)' % (X[i].tolist(), predictions[i], y[i]))
```

III.Les difficultés

- Choisir les bonnes données.
- Faire du tri dans les données.
- Créer un réseau de neurones.

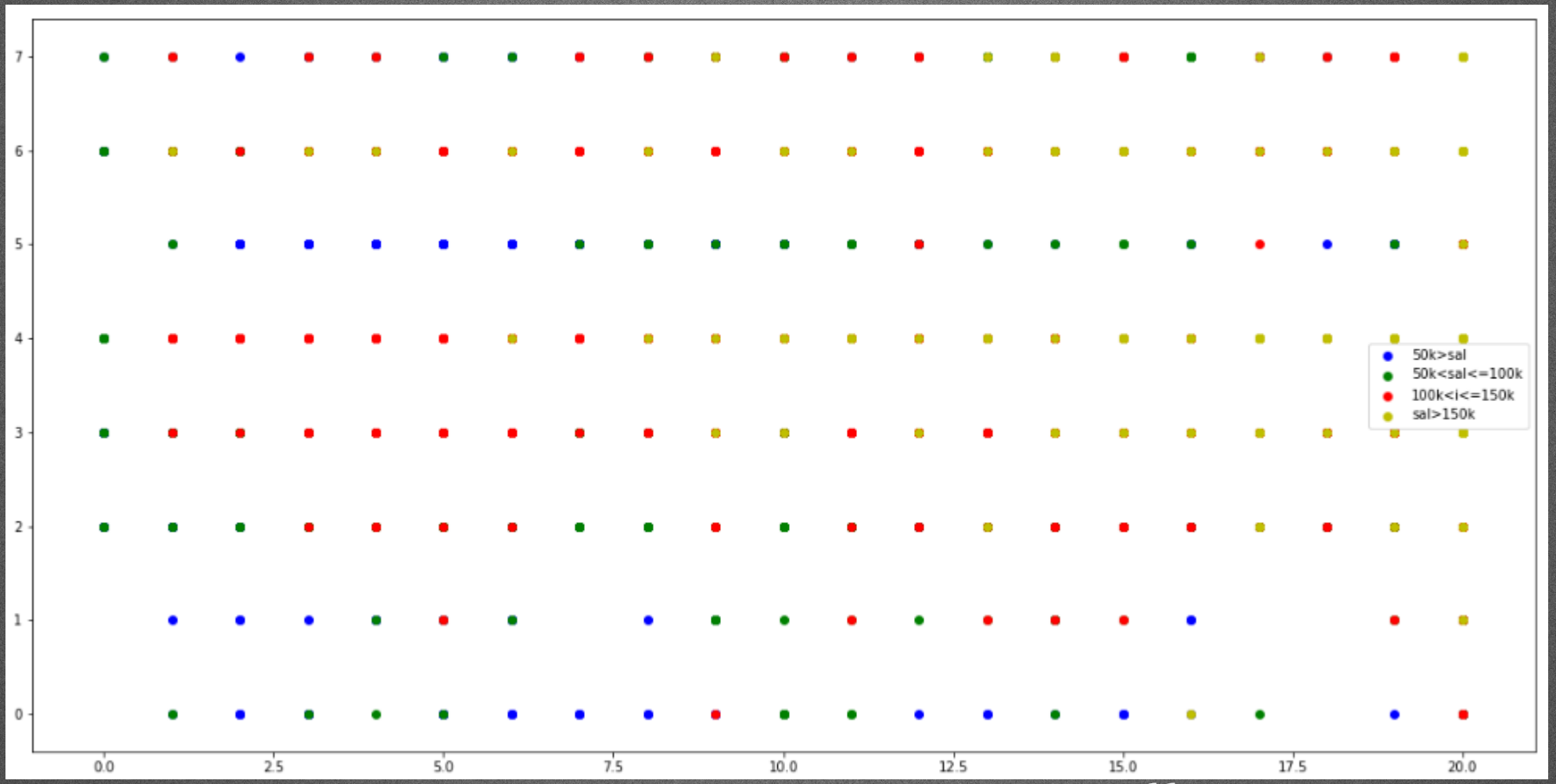
IV.Les résultats

· Après avoir trié les données, nous n’avons gardé que quelques colonnes

·Malgré nos essais, le réseau de neurones n’a pas fourni de résultats satisfaisants

·Corrélation entre certains critères et le salaire

Formal Education



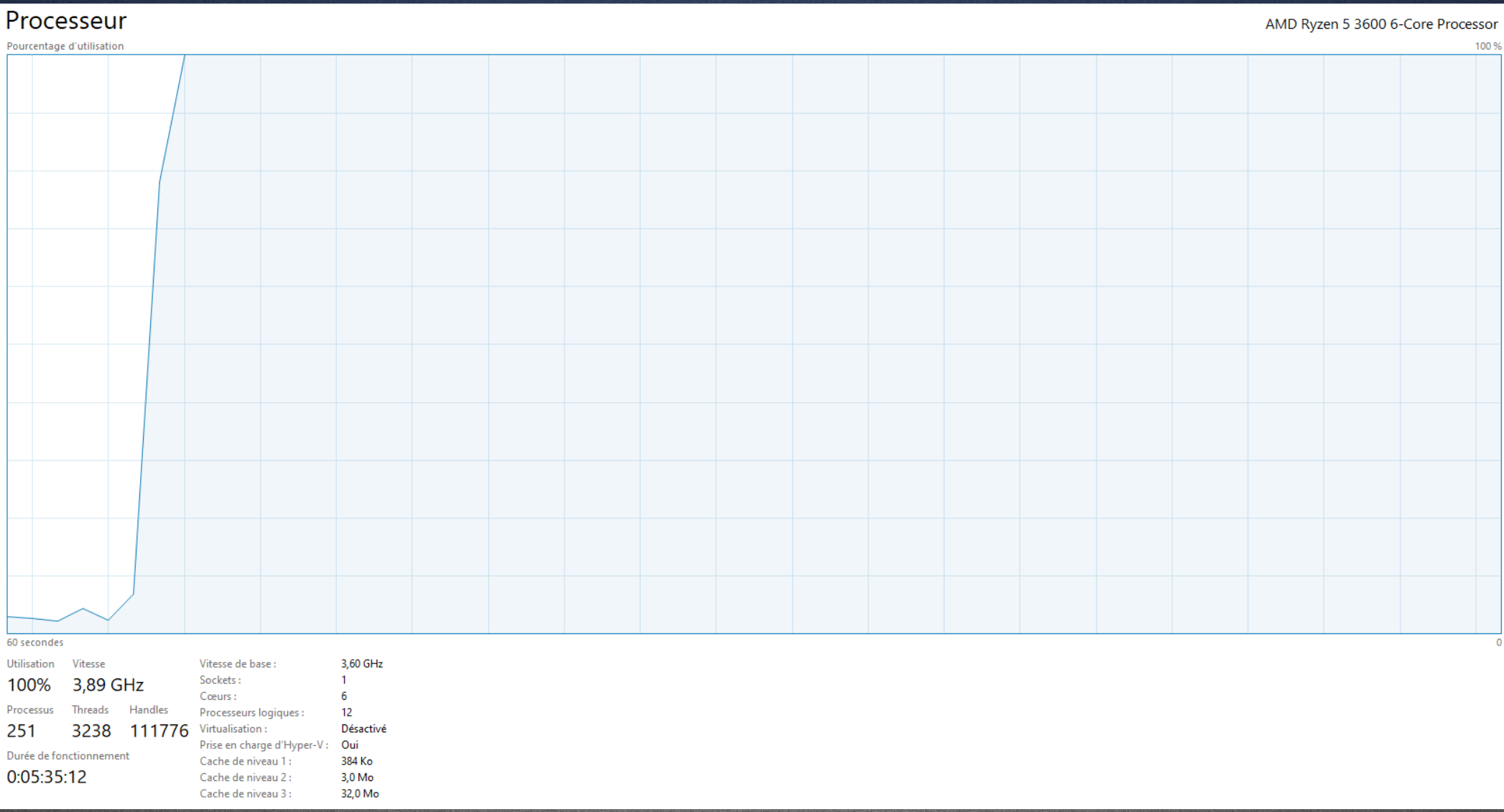
Years

```
dataFinal=dataFinal.replace("I never completed any formal education",0 )
dataFinal=dataFinal.replace("Primary/elementary school",1 )
dataFinal=dataFinal.replace("Secondary school",2 )
dataFinal=dataFinal.replace("Some college/university study without earning a bachelor's degree",3 )
dataFinal=dataFinal.replace("Bachelor's degree",4 )
dataFinal=dataFinal.replace("Professional degree",5 )
dataFinal=dataFinal.replace("Master's degree",6 )
dataFinal=dataFinal.replace("Doctoral degree",7 )
dataFinal=dataFinal.replace("I prefer not to answer",np.nan )
```

0 : Pas études formelles
1 : École primaire
2 : S’est arrêté au lycée
3 : N'a pas le bac mais a fini lycée
4 : A le bac
5 : A un bac pro
6 : Master
7 : Doctorant

```
Epoch 100/100
12834/12834 [==============================] - 1s 54us/step - loss: 0.3713 - accuracy: 0.8526
12834/12834 [==============================] - 0s 17us/step
Accuracy: 85.26
[20.0, 4.0] => 1 (expected 0)
[20.0, 5.0] => 1 (expected 1)
[20.0, 4.0] => 1 (expected 0)
[2.0, 4.0] => 1 (expected 1)
[10.0, 4.0] => 1 (expected 0)
[20.0, 6.0] => 1 (expected 0)
[7.0, 4.0] => 1 (expected 1)
[20.0, 3.0] => 1 (expected 1)
[20.0, 3.0] => 1 (expected 0)
[4.0, 4.0] => 1 (expected 1)
[8.0, 6.0] => 1 (expected 1)
[11.0, 6.0] => 1 (expected 1)
[3.0, 4.0] => 1 (expected 1)
[11.0, 6.0] => 1 (expected 1)
[5.0, 4.0] => 1 (expected 1)
[9.0, 4.0] => 1 (expected 1)
[2.0, 3.0] => 1 (expected 1)
[20.0, 3.0] => 1 (expected 0)
[20.0, 6.0] => 1 (expected 1)
[20.0, 3.0] => 1 (expected 1)
```

```
: print(dataFinal.catSalBin.sum()/12834)
0.8525790868006857
```


Processeur
AMD Ryzen 5 3600 6-Core Processor
Pourcentage d'utilisation
100 %
60 secondes
0
Utilisation
100%
Vitesse
3,89 GHz
Processus
251
Threads
3238
Handles
111776
Durée de fonctionnement
0:05:35:12
Vitesse de base : 3,60 GHz
Sockets : 1
Cœurs : 6
Processeurs logiques : 12
Virtualisation : Désactivé
Prise en charge d'Hyper-V : Oui
Cache de niveau 1 : 384 Ko
Cache de niveau 2 : 3,0 Mo
Cache de niveau 3 : 32,0 Mo

Les pistes d’améliorations

·Avoir un sondage avec des questions plus pertinentes pour avoir plus de données exploitables.

·Réussir à programmer un réseau de neurones pouvant prévoir le salaire.

Questions?