Prédire la souscription des clients à un produit bancaire

- Keras classifier

- Lahlou BENIDIRI

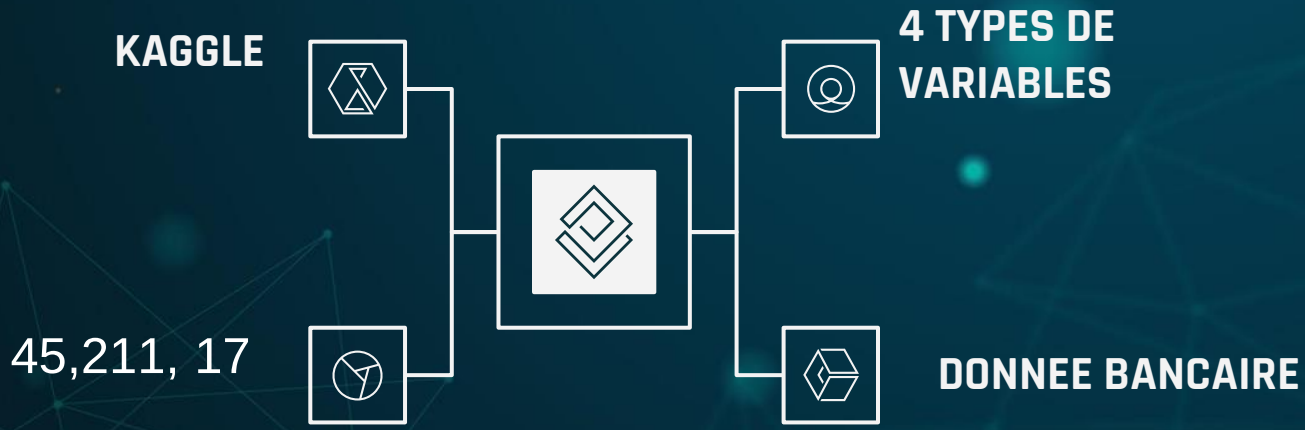
PLAN DE PRESENTATION

1- CONTEXTE ET PROBLÉMATIQUE
2- CHOIX DU DATA SET
3- PRÉ-TRAITEMENT ET ANALYSE DE DONNÉES
4- MODELISATION ET ARCHITECTURE
5- RESULTAT

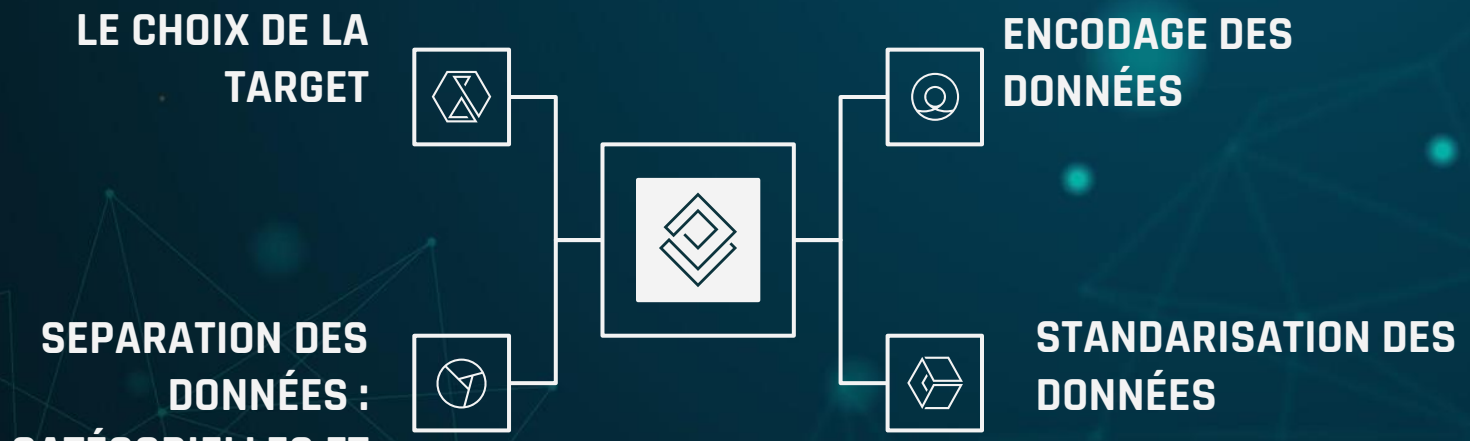
PROBLÉMATIQUE

Comment prédire une souscription d'un client à un dépôt à terme ?

CHOIX DU DATA SET



PRÉ-TRAITEMENT ET ANALYSE DE DONNÉES



Jeux de donnée Bank full

age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

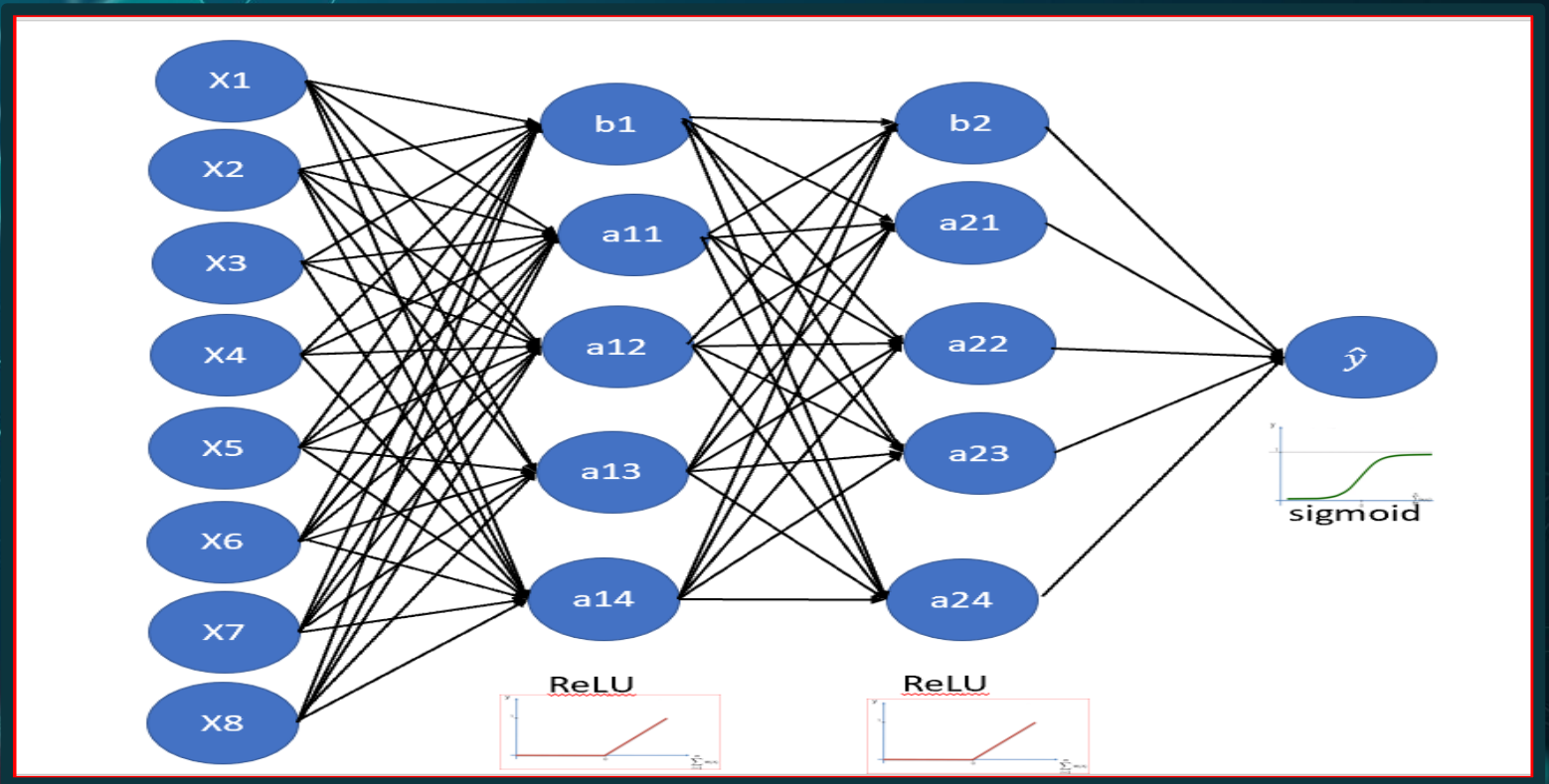
Source

kaggle

Correlation entre les variables

	age	balance	day	duration	campaign	pdays	previous
age	1.000000	0.097783	-0.009120	-0.004648	0.004760	-0.023758	0.001288
balance	0.097783	1.000000	0.004503	0.021560	-0.014578	0.003435	0.016674
day	-0.009120	0.004503	1.000000	-0.030206	0.162490	-0.093044	-0.051710
duration	-0.004648	0.021560	-0.030206	1.000000	-0.084570	-0.001565	0.001203
campaign	0.004760	-0.014578	0.162490	-0.084570	1.000000	-0.088628	-0.032855
pdays	-0.023758	0.003435	-0.093044	-0.001565	-0.088628	1.000000	0.454820
previous	0.001288	0.016674	-0.051710	0.001203	-0.032855	0.454820	1.000000

Architecture d'un model neuronal



ARCHITECTURE DU MODELE

```
classifier=Sequential()
```

```
classifier.add(Dense(units=25,activation ="relu", kernel_initializer ="uniform",input_dim=51))
classifier.add(Dropout(rate=0.1))
```

```
classifier.add(Dense(units=25,activation="relu",kernel_initializer="uniform"))
classifier.add(Dropout(rate=0.1))
```

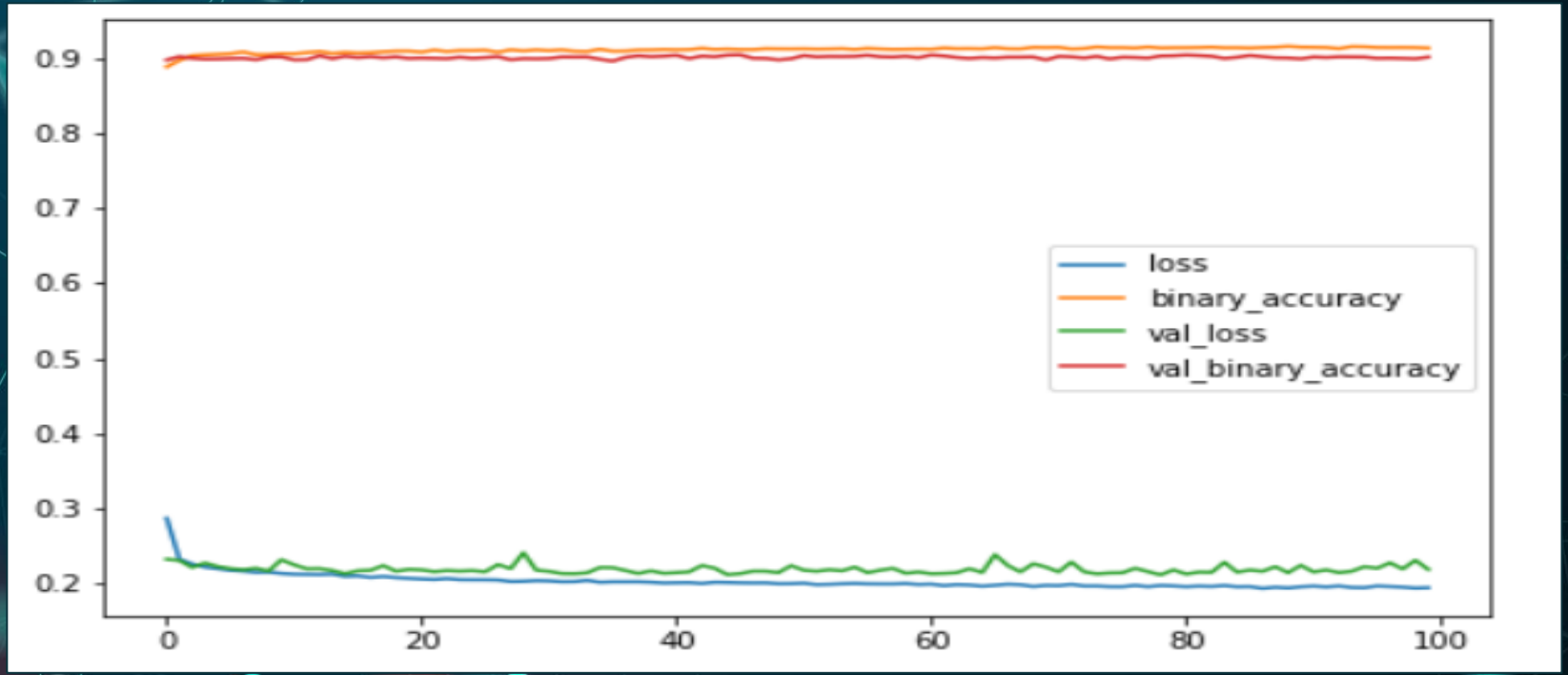
```
classifier.add(Dense(units=1, activation ="sigmoid",kernel_initializer="uniform"))
```

Entrainement

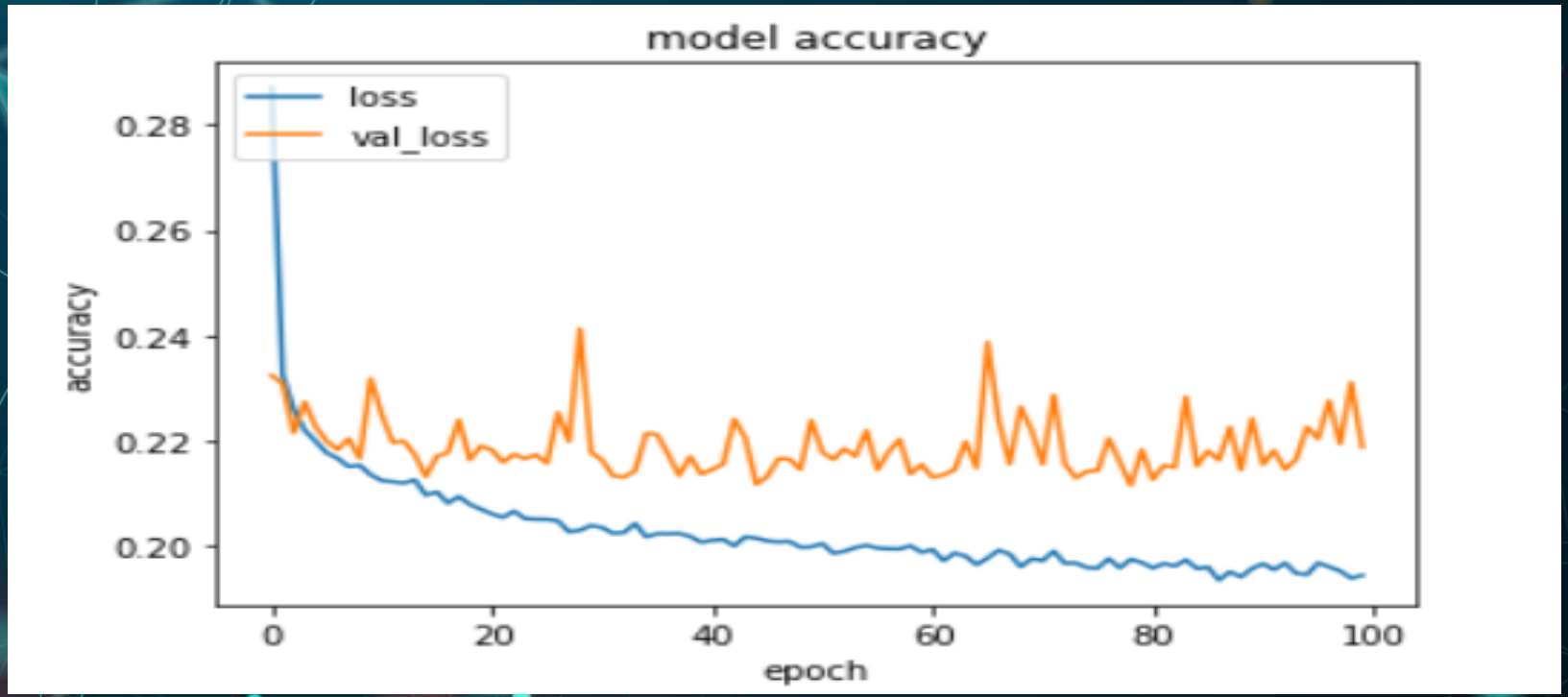
Layer (type)	Output Shape	Param #
dense_6 (Dense)	(None, 25)	1300
dropout_5 (Dropout)	(None, 25)	0
dense_7 (Dense)	(None, 25)	650
dropout_6 (Dropout)	(None, 25)	0
dense_8 (Dense)	(None, 25)	650
dropout_7 (Dropout)	(None, 25)	0
dense_9 (Dense)	(None, 25)	650
dropout_8 (Dropout)	(None, 25)	0
dense_10 (Dense)	(None, 1)	26

Total params: 3,276
Trainable params: 3,276
Non-trainable params: 0

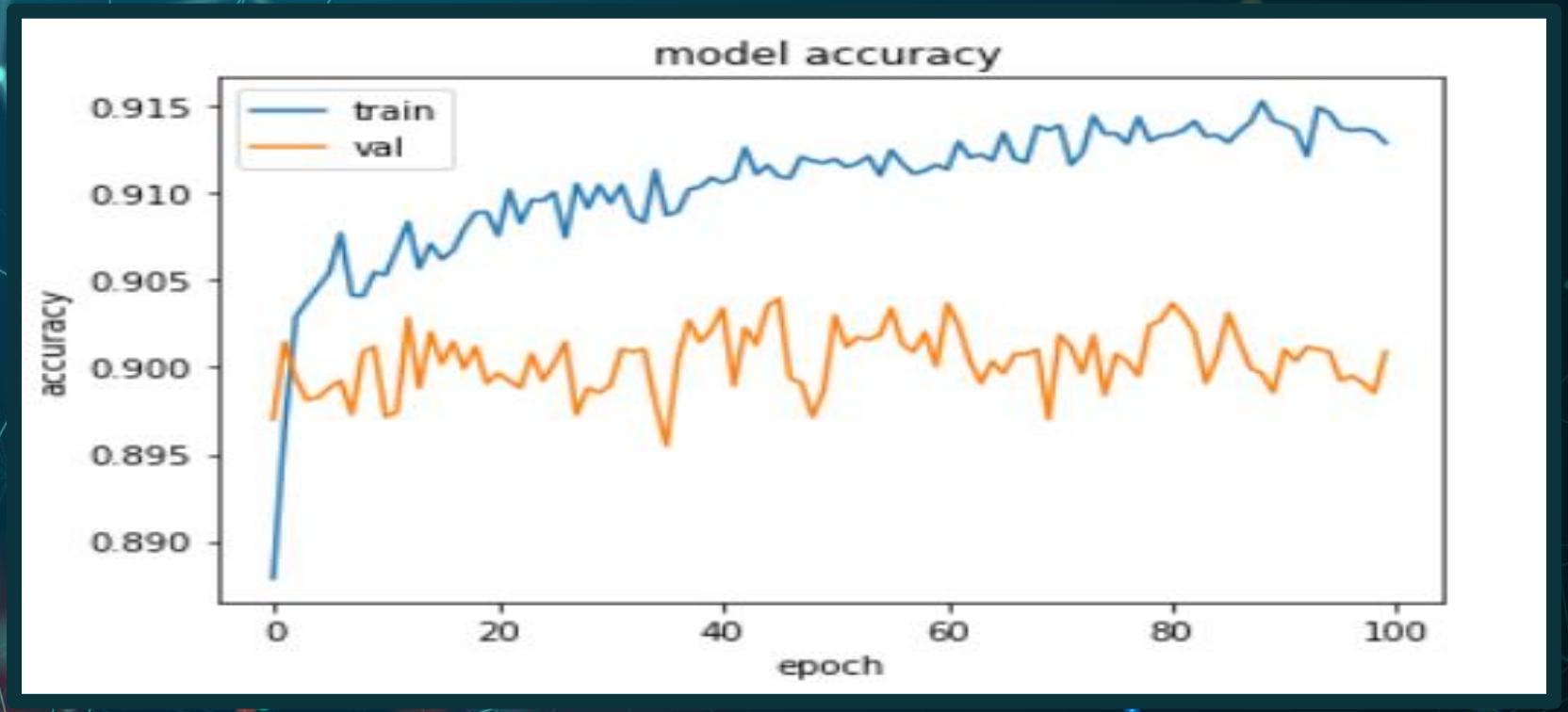
Graphique des métrique d'évaluation



Zoom sur loss



Zoom sur accuracy



=======>

Résultats

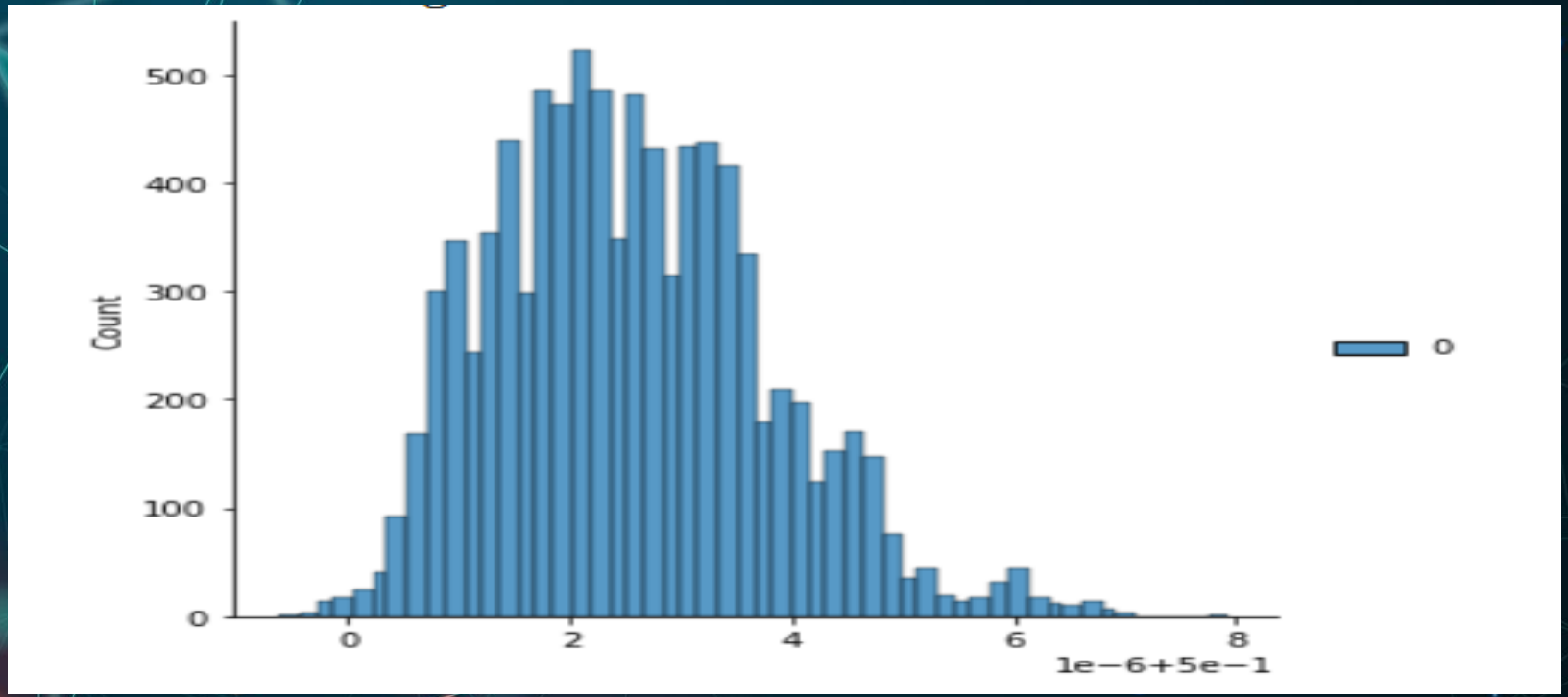
Efficacité 90,16%, Loss =0.2

Prediction

```
out_of_sample_predictions = classifier.predict(x_val)
out_of_sample_predictions
```

```
array([[0.50000167],
       [0.5000009 ],
       [0.50000465],
       ...,
       [0.5000018 ],
       [0.5000028 ],
       [0.50000155]], dtype=float32)
```

Representation graphique des predictions



Evaluation des predictions

```
def predict(j):
    ZZ= []
    z = classifier.predict(x_val)
    z[j]
    # si z est supérieur à 0,5, notre prédiction est de signe un (y_head=1),
    # si z est inférieur à 0,5, notre prédiction est de signe zéro (y_head=0),
    if z[j] <= 0.5:
        z = 0
        ZZ.append(z)
    else:
        z = 1
        ZZ.append(z)
    return ZZ
```

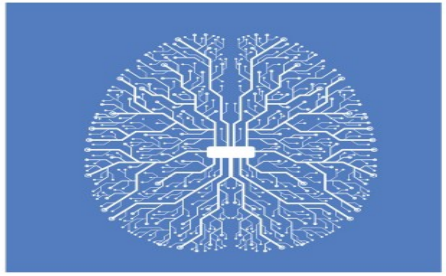
Comparaison avec le modèle de régression logistique

```
y_pred = logreg.predict(x_val)
print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(x_val, y_val)))
```

```
Accuracy of logistic regression classifier on test set: 0.90
```

Flask pour la mise en production

+



Resumé

1)Le modèle classifier avec 5 couches de keras est bon pour notre problématique malgrés la mauvaise corrélation entre les variables.

2) On peut dire que cette approche peut répondre à notre problématique de classification.

3)Flask est un bon outil pour la mise en production des modèles deep Learning.

4) notre modèle n'est pas mal par rapport à la régression logistique

THANKS!