INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

APPRENTISSAGE

TP

Réalisé par:

LAMAH Richard

Note: 8.5

Soumana Hamadou Abdourahmane

Note: 8

Introduction

Dataset: https://www.kaggle.com/uciml/sms-spam-collection-dataset

Les Algorithmes utilisés

- **RNN**
- **NAIVE BAYES**
- **NLP AND RAMDOM FOREST**

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Importation des bibliothèques nécessaires

Importation du Dataset

```
dataset=pd.read_csv("spam.csv",delimiter=',',encoding='latin-1')
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
5	spam	FreeMsg Hey there darling it's been 3 week's n...	NaN	NaN	NaN

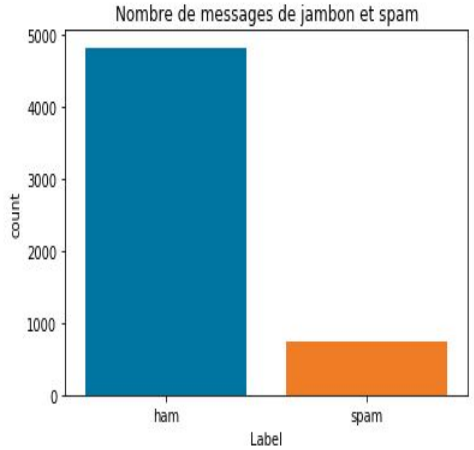
Supprimer les colonnes non requises pour le réseau de neurones

```
dataset.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],axis=1,inplace=True)
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Comprendre mieux la distribution des données.

```
Text(0.5, 1.0, 'Nombre de messages de jambon et spam')
```

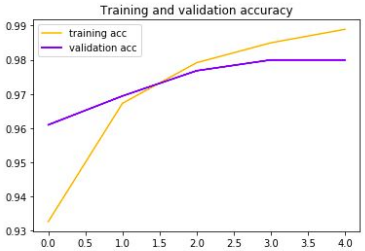


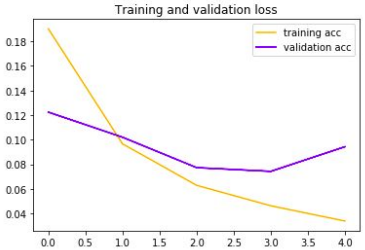

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inputs (InputLayer)          (None, 150)               0
_________________________________________________________________
embedding_1 (Embedding)      (None, 150, 50)           50000
_________________________________________________________________
lstm_1 (LSTM)                (None, 64)                29440
_________________________________________________________________
FC1 (Dense)                  (None, 256)               16640
_________________________________________________________________
activation_1 (Activation)    (None, 256)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
out_layer (Dense)            (None, 1)                 257
_________________________________________________________________
activation_2 (Activation)    (None, 1)                 0
=================================================================
Total params: 96,337
Trainable params: 96,337
Non-trainable params: 0
```

```
Epoch 4/10
3788/3788 [==============================] - 7s 2ms/step - loss: 0.0463 - accuracy: 0.9850 - val_loss: 0.0743 - va
l_accuracy: 0.9800
Epoch 5/10
3788/3788 [==============================] - 8s 2ms/step - loss: 0.0339 - accuracy: 0.9889 - val_loss: 0.0943 - va
l_accuracy: 0.9800
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Interpretation graphique






```
test_sequences = tok.texts_to_sequences(X_test)
test_sequences_matrix = sequence.pad_sequences(test_sequences,maxlen=max_len)
```

```
accr = model.evaluate(test_sequences_matrix,Y_test)
```

```
836/836 [==============================] - 1s 863us/step
```

```
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
Test set
  Loss: 0.103
  Accuracy: 0.976
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

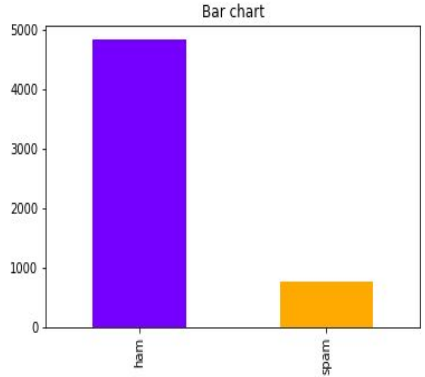
Importation du Dataset

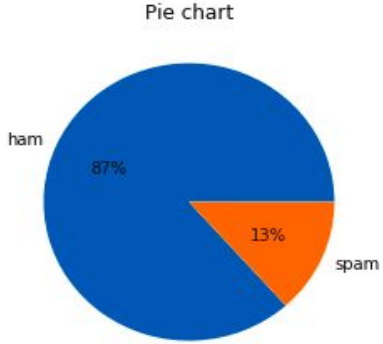
	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
5	spam	FreeMsg Hey there darling it's been 3 week's n...	NaN	NaN	NaN
6	ham	Even my brother is not like to speak with me. ...	NaN	NaN	NaN
7	ham	As per your request 'Melle Melle (Oru Minnamin...	NaN	NaN	NaN
8	spam	WINNER!! As a valued network customer you have...	NaN	NaN	NaN
9	spam	Had your mobile 11 months or more? U R entitle...	NaN	NaN	NaN

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

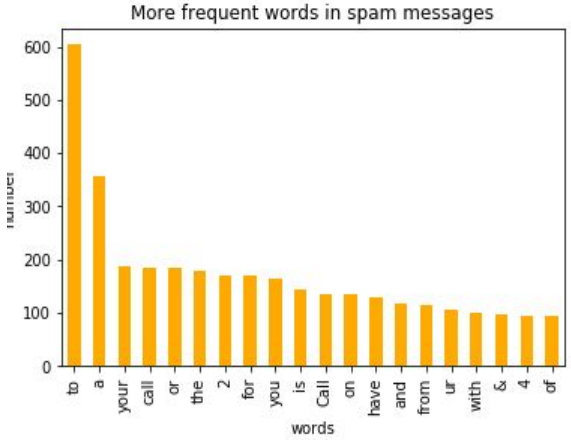
Les résultats graphique (Distribution spam/non-spam plots)





Analyse de texte

Nous voulons trouver la fréquence des mots dans les spams et les non-spams. Les mots des messages seront des caractéristiques du modèle.Nous utilisons la fonction Counter.



Naive bayes

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Nous pouvons voir que la majorité des mots fréquents dans les deux classes sont des mots vides tels que "to", "a", "or" et ainsi de suite.
Avec les mots vides, nous nous référons aux mots les plus courants d'une langue, il n'y a pas de liste simple et universelle de mots vides.
Ingénierie des fonctionnalités
Le prétraitement du texte, la segmentation et le filtrage des mots vides sont inclus dans un composant de haut niveau capable de créer un dictionnaire de caractéristiques et de transformer des documents en vecteurs de caractéristiques.
Nous supprimons les mots vides afin d'améliorer l'analyse

Analyse prédictive
Notre objectif est de prédire si un nouveau SMS est du spam ou non. On suppose qu'il est bien pire de classer incorrectement un non-spam que de classer incorrectement un spam. (on ne veux pas avoir de faux positifs)

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

Les résultats graphique

	alpha	Train Accuracy	Test Accuracy	Test Recall	Test Precision
0	0.00001	0.998661	0.974443	0.920635	0.895753
1	0.11001	0.997857	0.976074	0.936508	0.893939
2	0.22001	0.997857	0.977162	0.936508	0.900763
3	0.33001	0.997589	0.977162	0.936508	0.900763
4	0.44001	0.997053	0.977162	0.936508	0.900763
5	0.55001	0.996250	0.976618	0.936508	0.897338
6	0.66001	0.996518	0.976074	0.932540	0.896947
7	0.77001	0.996518	0.976074	0.924603	0.903101
8	0.88001	0.996250	0.976074	0.924603	0.903101
9	0.99001	0.995982	0.976074	0.920635	0.906250

```
best_index = models['Test Precision'].idxmax()
models.iloc[best_index, :]
```

```
alpha             15.730010
Train Accuracy     0.979641
Test Accuracy      0.969549
Test Recall        0.777778
Test Precision     1.000000
Name: 143, dtype: float64
```

	alpha	Train Accuracy	Test Accuracy	Test Recall	Test Precision
143	15.73001	0.979641	0.969549	0.777778	1.0
144	15.84001	0.979641	0.969549	0.777778	1.0
145	15.95001	0.979641	0.969549	0.777778	1.0
146	16.06001	0.979373	0.969549	0.777778	1.0
147	16.17001	0.979373	0.969549	0.777778	1.0

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

Les résultats precision

```
best_index = models[models['Test Precision']==1]['Test Accuracy'].idxmax()
bayes = naive_bayes.MultinomialNB(alpha=list_alpha[best_index])
bayes.fit(X_train, y_train)
models.iloc[best_index, :]
```

```
alpha             15.730010
Train Accuracy     0.979641
Test Accuracy      0.969549
Test Recall        0.777778
Test Precision     1.000000
Name: 143, dtype: float64
```

```
m_confusion_test = metrics.confusion_matrix(y_test, bayes.predict(X_test))
pd.DataFrame(data = m_confusion_test, columns = ['Predicted 0', 'Predicted 1'],
             index = ['Actual 0', 'Actual 1'])
```

	Predicted 0	Predicted 1
Actual 0	1587	0
Actual 1	56	196

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

Les 10 premiers rang

	C	Train Accuracy	Test Accuracy	Test Recall	Test Precision
0	500.0	0.994910	0.982599	0.873016	1.0
1	600.0	0.995982	0.982599	0.873016	1.0
2	700.0	0.996785	0.982599	0.873016	1.0
3	800.0	0.997053	0.983143	0.876984	1.0
4	900.0	0.997589	0.983143	0.876984	1.0
5	1000.0	0.998125	0.983143	0.876984	1.0
6	1100.0	0.998928	0.983143	0.876984	1.0
7	1200.0	0.999732	0.983143	0.876984	1.0
8	1300.0	1.000000	0.983143	0.876984	1.0
9	1400.0	1.000000	0.983143	0.876984	1.0

```
models[models['Test Precision']==1].head(n=5)
```

	C	Train Accuracy	Test Accuracy	Test Recall	Test Precision
0	500.0	0.994910	0.982599	0.873016	1.0
1	600.0	0.995982	0.982599	0.873016	1.0
2	700.0	0.996785	0.982599	0.873016	1.0
3	800.0	0.997053	0.983143	0.876984	1.0
4	900.0	0.997589	0.983143	0.876984	1.0

```
C                 500.000000
Train Accuracy      0.994910
Test Accuracy       0.982599
Test Recall         0.873016
Test Precision      1.000000
Name: 0, dtype: float64
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Naive bayes

```
best_index = models[models['Test Precision']==1]['Test Accuracy'].idxmax()
svc = svm.SVC(C=list_C[best_index])
svc.fit(X_train, y_train)
models.iloc[best_index, :]
```

```
C                  800.000000
Train Accuracy       0.997053
Test Accuracy        0.983143
Test Recall          0.876984
Test Precision       1.000000
Name: 3, dtype: float64
```

```
m_confusion_test = metrics.confusion_matrix(y_test, svc.predict(X_test))
pd.DataFrame(data = m_confusion_test, columns = ['Predicted 0', 'Predicted 1'],
             index = ['Actual 0', 'Actual 1'])
```

	Predicted 0	Predicted 1
Actual 0	1587	0
Actual 1	31	221

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

NLP and Ramdom forest

Importer des données et comprendre des données

```
sample messages from human
0     Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wa
t...
1     Ok lar... Joking wif u oni...
3     U dun say so early hor... U c already then say...
4     Nah I don't think he goes to usf, he lives around here though
6     Even my brother is not like to speak with me. They treat me like aids patent.
7     As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for a
ll Callers. Press *9 to copy your friends Callertune
10    I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough toda
y.
13    I've been searching for the right words to thank you for this breather. I promise i wont take your help for
granted and will fulfil my promise. You have been wonderful and a blessing at all times.
14    I HAVE A DATE ON SUNDAY WITH WILL!!
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

NLP and Ramdom forest

Écrire une fonction propre

```
#Write a clean function
import string
import nltk
#ps = nltk.PorterStemmer()
stopwords= nltk.corpus.stopwords.words('english')

def clean(sentence):
    s = "".join(x for x in sentence if x not in string.punctuation)
    temp = s.lower().split(' ')
    temp2 = [x for x in temp if x not in stopwords]
    return temp2
clean("hell peOople  are hOOow ! AAare ! you. enough.. are")
```

```
['hell', 'peoople', '', 'hooow', '', 'aaare', '', 'enough']
```

Créer un vectoriseur et transformer en entités de colonne

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

NLP and Ramdom forest

Créer un vectoriseur et transformer en entités de colonne

```
print (vector_output [0:10])
```

```
  (0, 3750)      0.15133352947689135
  (0, 4646)      0.3328581634691494
  (0, 6378)      0.26034665364508613
  (0, 2465)      0.2577757346943601
  (0, 1377)      0.2531117123219365
  (0, 1842)      0.28119992157437884
  (0, 5626)      0.18060020249787317
  (0, 3847)      0.18526422487029676
  (0, 9110)      0.2295416796796082
  (0, 4805)      0.28119992157437884
  (0, 2990)      0.197308809001304
  (0, 1840)      0.3177490962883377
```

5569	0.154050	0.0	0.0	0.0	0.0	0.0	0.0
5570	0.000000	0.0	0.0	0.0	0.0	0.0	0.0
5571	0.000000	0.0	0.0	0.0	0.0	0.0	0.0

5572 rows × 9434 columns

```
pd.DataFrame(vector_output.toarray())
```

	0	1	2	3	4	5	6	7	8	9	...	9424	9425	9426	9427	9428	9429	9430	9431	9432	9433
0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0
5	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.0

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

NLP and Ramdom forest

Créer x caractéristiques et fractionner les données lors des tests et de la formation

```
[(0.15051216146506421, 'short_number'),
 (0.06745598099009252, 'long_number'),
 (0.03780295645809816, 'len'),
 (0.023755114809081462, 1908),
 (0.0222565233851357, 8520),
 (0.021854327382716452, 3541),
 (0.019653275501129913, 5472),
 (0.014622683943416943, 2186),
 (0.012011149107612317, 7250),
 (0.011653897338884453, 6540),
 (0.011000307905711756, 7809),
 (0.009596361397345859, 8152),
 (0.008051423679147799, 6881),
 (0.00736589449306907, 382),
 (0.006732281182267652, 8646),
 (0.0066570320371043224, 2005),
 (0.006480591202430112, 5791),
 (0.006149119340504071, 9020),
 (0.005874207038055217, 9372),
 (0.0057352039034421565, 3885)]
```

Prédire et vérifier votre score

```
Precision : 0.988 / Recall : 0.906 / fscore : 0.945 / Accuracy: 0.986
```

INSTITUT
DE LA FRANCOPHONIE
POUR L'INNOVATION

Conclusion

Algorithme RNN

Train_set

```
  accuracy: 0.9863
Test set
  Loss: 0.103
  Accuracy: 0.976
```

Algorithme Naive bayes

```
C                800.000000
Train Accuracy     0.997053
Test Accuracy      0.983143
Test Recall        0.876984
Test Precision     1.000000
Name: 3, dtype: float64
```

Algorithme NLP AND RAMDOM FOREST

```
Precision : 0.988 / Recall : 0.906 / fscore : 0.945 / Accuracy: 0.986
```