Project report

Step 1 : data importation and label encoding :

results

	sent	y
0	'طوال حياتي لم المس اي تغير حتى قدمت هذه الحكو...	1
1	'منتوج رائع وثمن مناسب ....جميل'	1
2	'كلنا ابن كيران لمتافق معايا يدير جيم'	1
3	'وفقك الله لولاية اخرى حقّاش مكينش محسن منك'	1
4	'لأنه و بكل بساطة رئيس الحكومة يعتني بمعاق داخ...	1
...	...	...
1995	'اصمت لعلى صمتك راحة بالنسبة لهم'	0
1996	'حديقة حيوانات ولازال هنالك اناس لا يؤمنون بنظ...	0
1997	'أفعى بجدارة تريثت تربصت وكان الفحيح متعة له ص...	0
1998	'لا يقطع الرأس غير الي ركبه الان اصبح تركيب ال...	0
1999	'امة النون نستنكر ندين نشجب ثم نوافق'	0

2000 rows × 2 columns

Step 2 : data cleaning and preprocessing :

In this step I have removed diacritics , longations , english words, repetitions, spaces/tabulations/new lines ..., digits, and stopwords

Finally I used `ISRIStemmer` to stem the data

	sent	y
0	طول حيت لمس اي تغر حتي قدم حكم فل نقّف بجن بصت	1
1	نتج رءع وثم نسب جمل	1
2	كلن ابن كير تفق معا يدر	1
3	وفق الل لول اخر حقّش كينش حسن منك	1
4	لنه بكل بسط رءس حكم يعت عاق دخل بيت الل ميز حسن	1

Step3 : data splitting :

```
print(X_train.shape)
print(X_test.shape)
```

```
(1600,)
(400,)
```

Step3 : Modeling:

First model : TF-IDF + Support vector classifier

Accuracy score is 0.84

	precision	recall	f1-score	support
0	0.81	0.91	0.85	204
1	0.89	0.78	0.83	196
accuracy			0.84	400
macro avg	0.85	0.84	0.84	400
weighted avg	0.85	0.84	0.84	400

Second model : CNN

```
Epoch 1/5
50/50 - 5s - loss: 0.6666 - accuracy: 0.6294 - val_loss: 0.5866 - val_accuracy: 0.7800 - 5s/epoch - 97ms/step
Epoch 2/5
50/50 - 3s - loss: 0.3294 - accuracy: 0.8975 - val_loss: 0.3512 - val_accuracy: 0.8575 - 3s/epoch - 62ms/step
Epoch 3/5
50/50 - 3s - loss: 0.0898 - accuracy: 0.9719 - val_loss: 0.3942 - val_accuracy: 0.8450 - 3s/epoch - 61ms/step
Epoch 4/5
50/50 - 3s - loss: 0.0206 - accuracy: 0.9969 - val_loss: 0.4095 - val_accuracy: 0.8725 - 3s/epoch - 60ms/step
Epoch 5/5
50/50 - 3s - loss: 0.0069 - accuracy: 0.9994 - val_loss: 0.4705 - val_accuracy: 0.8500 - 3s/epoch - 61ms/step
13/13 - 0s - loss: 0.4705 - accuracy: 0.8500 - 58ms/epoch - 4ms/step
score: 0.47
acc: 0.85
```

Third model : LSTM

```
_________________________________________________________________
None
Epoch 1/10
50/50 - 14s - loss: 0.6481 - accuracy: 0.5025 - 14s/epoch - 276ms/step
Epoch 2/10
50/50 - 11s - loss: 0.4493 - accuracy: 0.5025 - 11s/epoch - 216ms/step
Epoch 3/10
50/50 - 11s - loss: 0.2071 - accuracy: 0.5025 - 11s/epoch - 217ms/step
Epoch 4/10
50/50 - 11s - loss: 0.1108 - accuracy: 0.5025 - 11s/epoch - 217ms/step
Epoch 5/10
50/50 - 11s - loss: 0.0756 - accuracy: 0.5025 - 11s/epoch - 219ms/step
Epoch 6/10
50/50 - 11s - loss: 0.0448 - accuracy: 0.5025 - 11s/epoch - 219ms/step
Epoch 7/10
50/50 - 11s - loss: 0.0324 - accuracy: 0.5025 - 11s/epoch - 217ms/step
Epoch 8/10
50/50 - 11s - loss: 0.0193 - accuracy: 0.5025 - 11s/epoch - 220ms/step
Epoch 9/10
50/50 - 11s - loss: 0.0150 - accuracy: 0.5025 - 11s/epoch - 222ms/step
Epoch 10/10
50/50 - 11s - loss: 0.0075 - accuracy: 0.5025 - 11s/epoch - 217ms/step
13/13 - 1s - loss: 0.6089 - accuracy: 0.4900 - 589ms/epoch - 45ms/step
score: 0.61
acc: 0.49
```

Comparaison

The LSTM result after the training and the validation step gave us a poor accuracy compared to the CNN, the model was even slower. Thus this can be improved by tuning the hyperparameter, it can be even faster as well if we combine it with another model including CNN itself.
SVC gave us some interesting results with 0.83 accuracy
CNN outperformed the 2 models with 0.85 even without hyperparameter tuning.