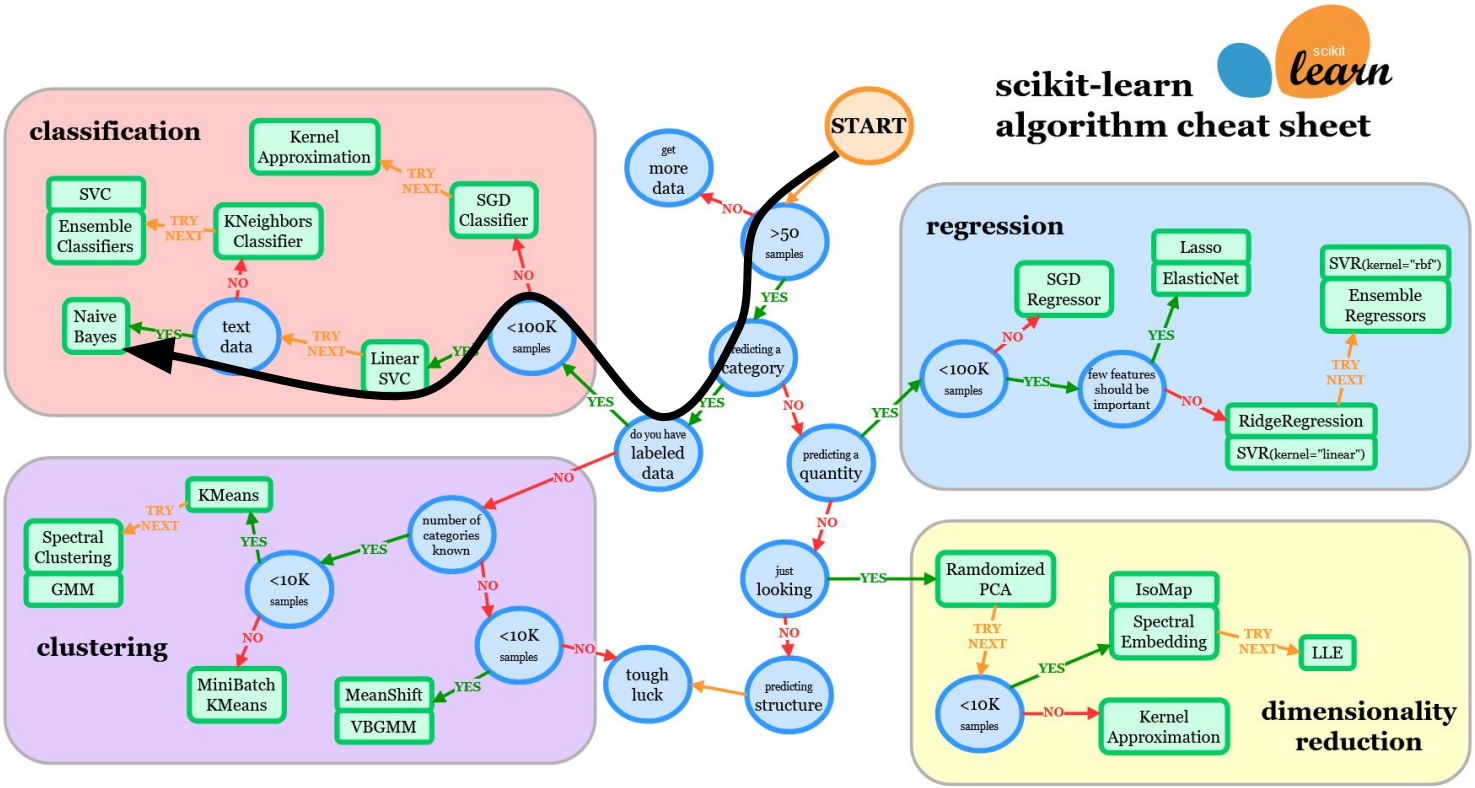
SpamClassifier

Fabien Herry
Melody Duplaix

Modèles testés



Premières comparaisons

LinearSVC Scikit-Learn après vectorisation

	precision	recall	f1-score	support
0	0.98	1.00	0.99	966
1	0.99	0.86	0.92	149
accuracy			0.98	1115
macro avg	0.99	0.93	0.95	1115
weighted avg	0.98	0.98	0.98	1115

	text	résult
0	Join us today ! Flexible work without constain...	ham
1	Chronopost : your package 7d6595466533 is wait...	ham
2	I'm a nigerian prince i have 250 millions doll...	spam
3	There was a hack on your amazon account clic o...	ham
4	Your energy provider want to talk to you about...	ham
5	URGENT: Your account has been compromised. Cal...	ham

Premières comparaisons

Naive Bayes simple

```
Exactitude : 0.92
Rappel : 0.99
Rappel négative: 0.65
Precision : 0.92
Précision négative: 0.93
F1-Score : 0.95
F1-Score Négatif: 0.76
```

	text	result
0	Join us today ! Flexible work without constain...	spam
1	Chronopost : your package 7d6595466533 is wait...	spam
2	I'm a nigerian prince i have 250 millions doll...	ham
3	There was a hack on your amazon account clic o...	ham
4	Your energy provider want to talk to you about...	ham
5	URGENT: Your account has been compromised. Cal...	spam

Pourtant, Modèle simple plus performant sur des spams écrit à la main, non dans le dataset

↪ Overfitting

Problématique exploratoire

Question de models:

- Quel modèle choisir
- Quel features
- Sur quel jeu de données entraîner

Question de Data :

- Chercher plus de données
- Définir un test de satisfaction

Automatisations

Trouver les combinaisons de features / modèles les plus efficaces

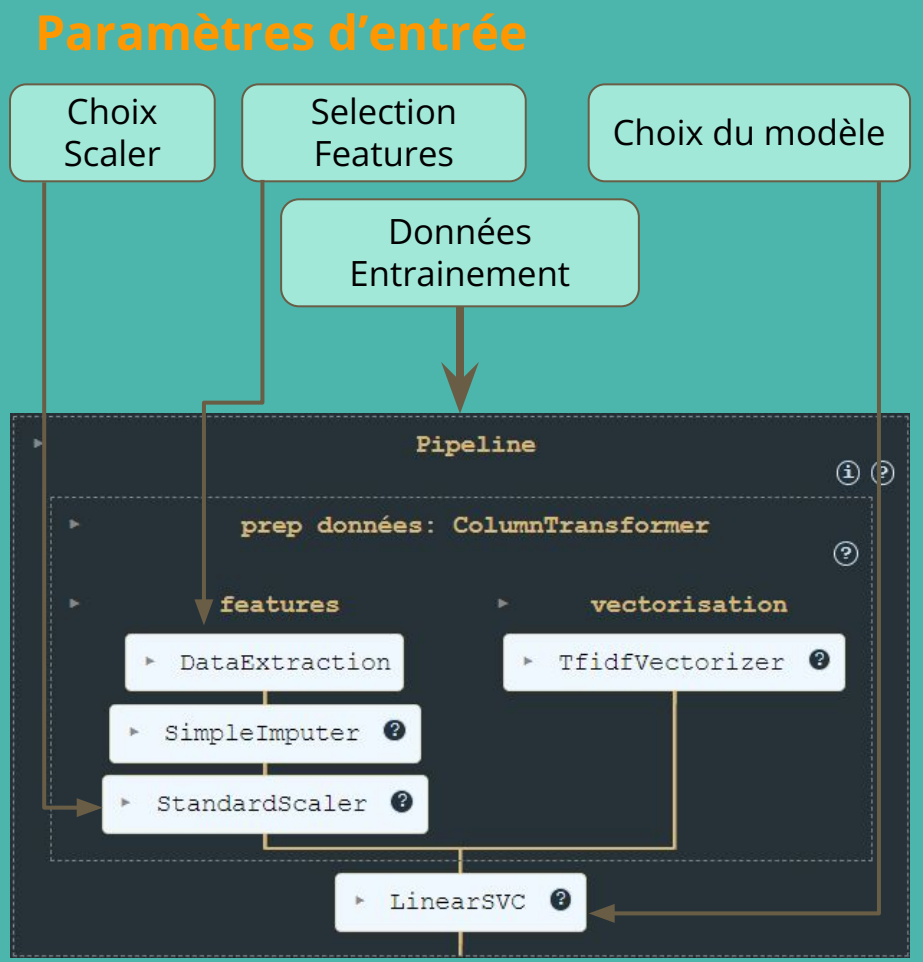
I - Module pour créer et entraîner un modèle différent rapidement avec le même préprocessing
→ pipeline

II - Automatisation d'une analyse de base de donnée

III - Modules pour tester l'efficacité de toutes les combinaisons de features possibles pour chaque modèle, pour chaque combinaison de dataset → validation croisée

Pipeline de base

Modulaire pour automatisation



II

Données disponible

-Base de données d'origine

-Base de données étudiant nigerien

-Telegram

-Email Enron . Probleme de formatage

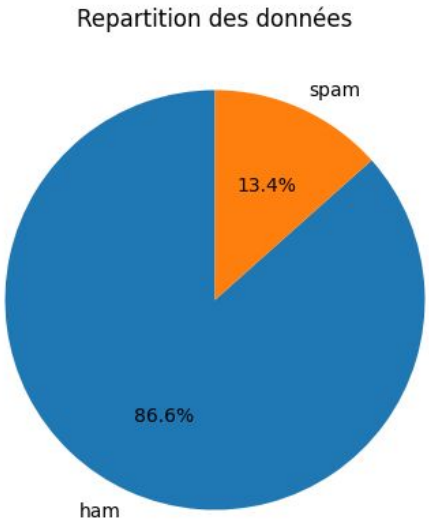
II

Exploration des données et features

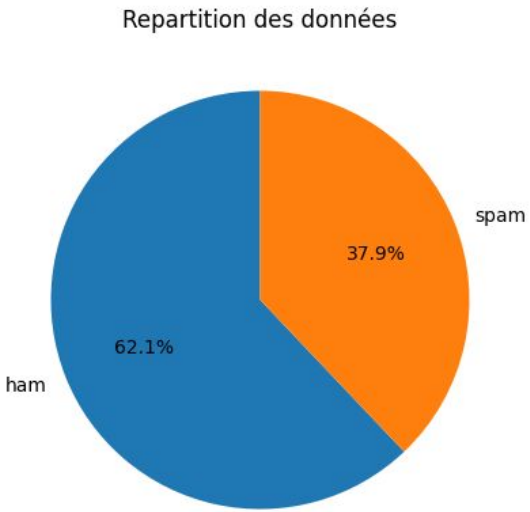
- Répartition spam / ham
- Nombre caractères par sms
- Nombre mots par sms
- Nombre de symboles monétaires
- Nombre de chiffres
- Nombre de liens

II

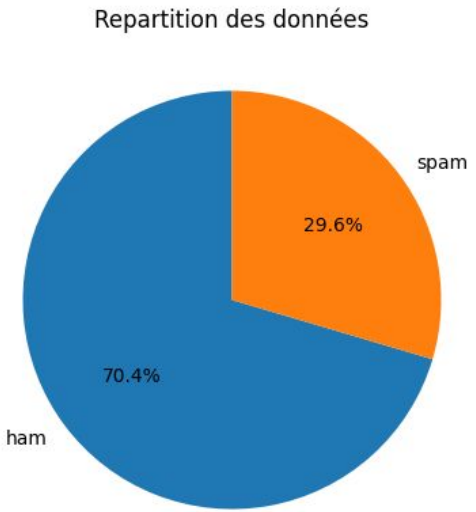
Visualisation des données



Dataset original



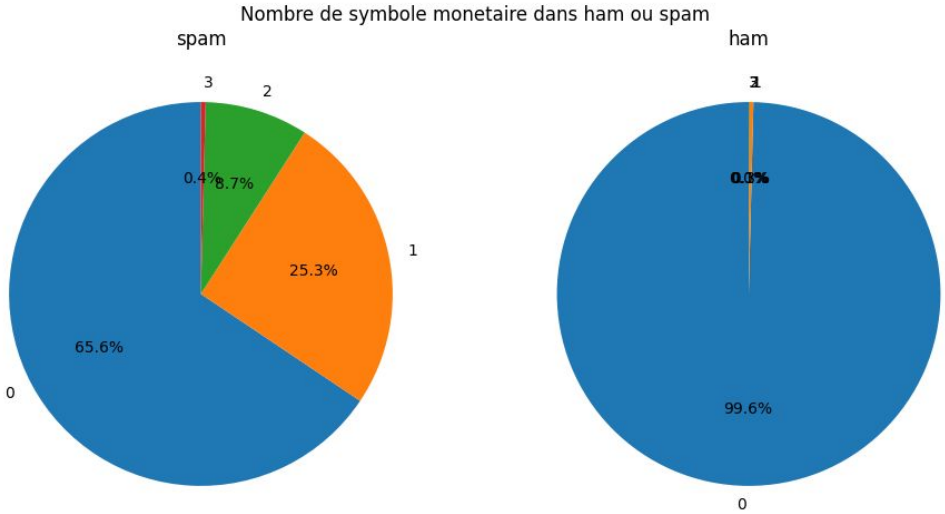
Dataset Nigérien



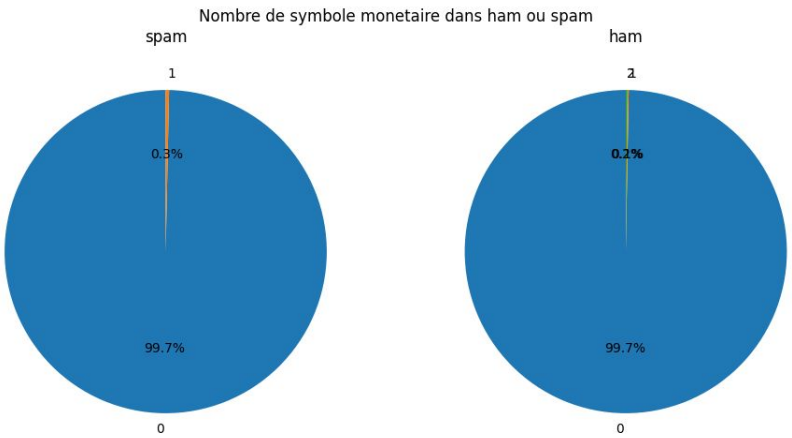
Dataset Télégram

II

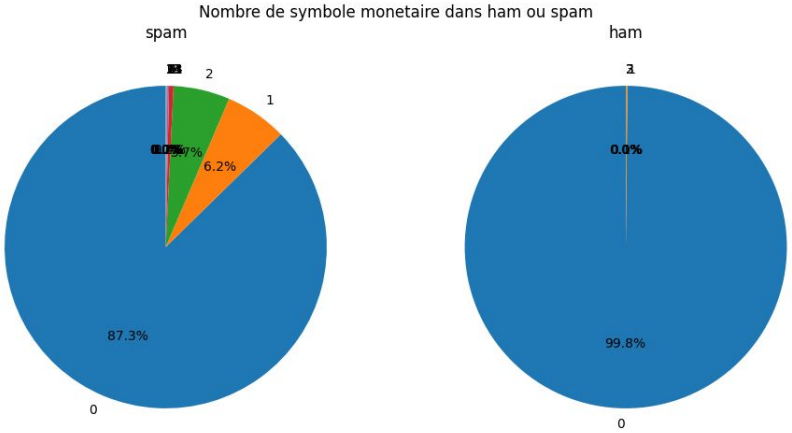
Visualisation des données



Dataset original



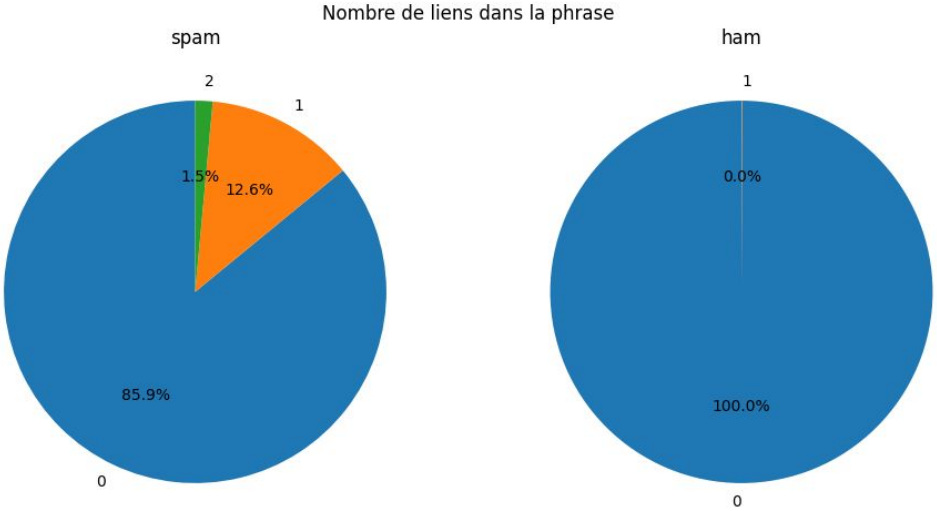
Dataset Nigérien



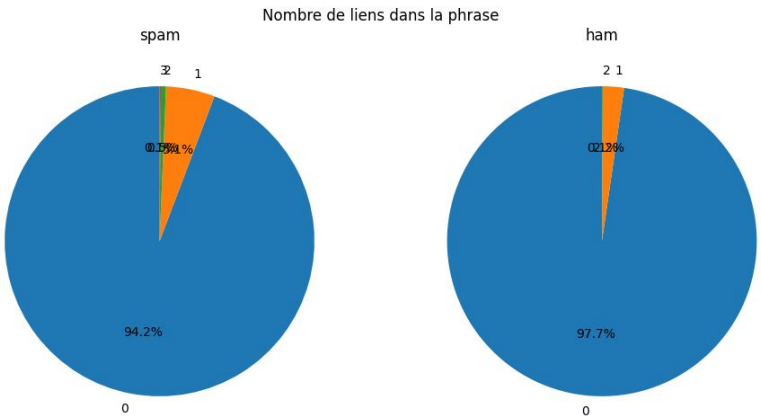
Dataset Télégram

II

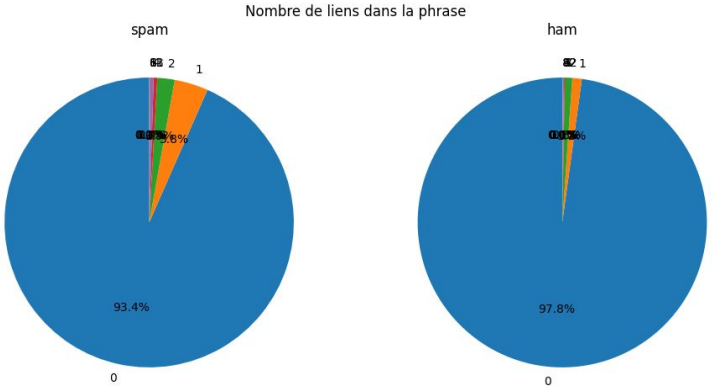
Visualisation des données



Dataset original



Dataset Nigérien

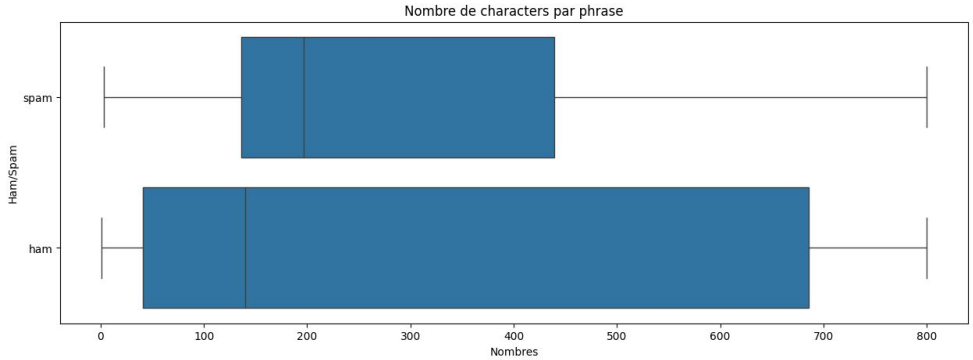


Dataset Télégram

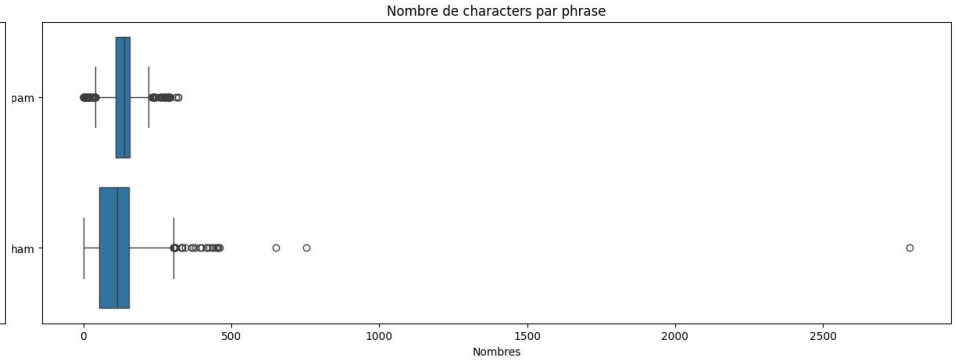
II

Visualisation des données

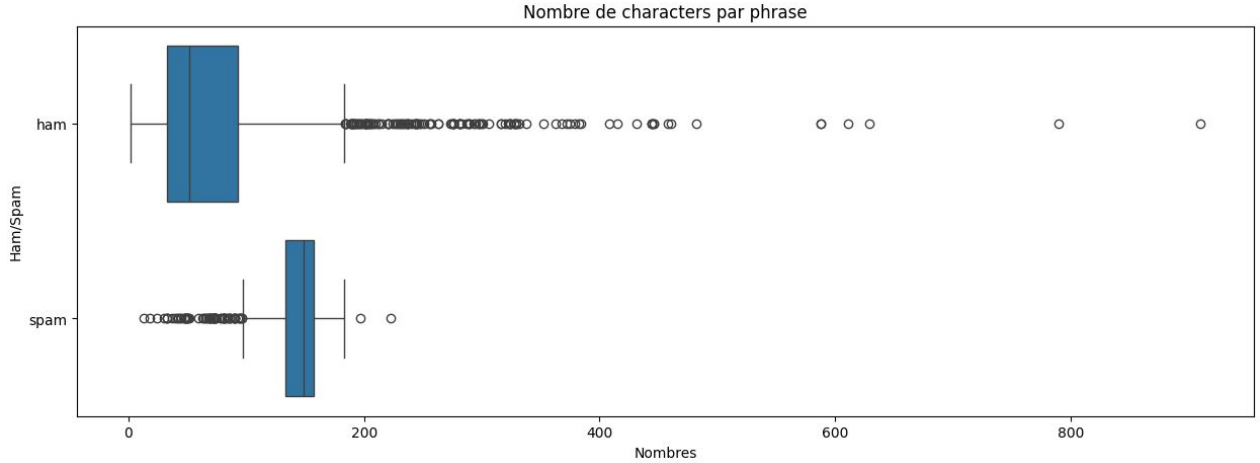
Dataset Niégerien



Dataset Télégram



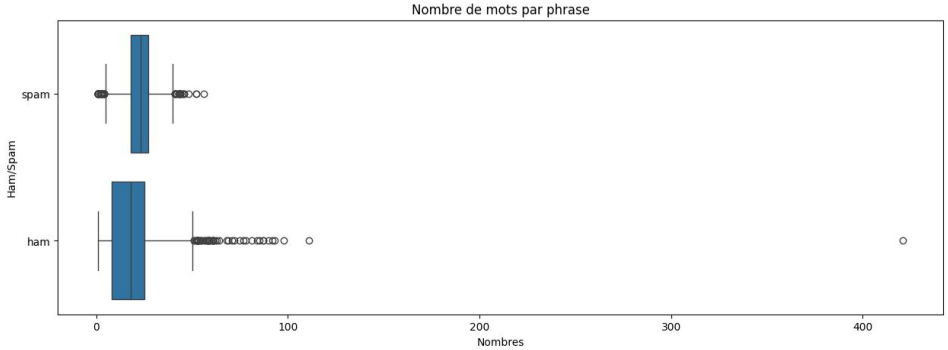
Dataset original



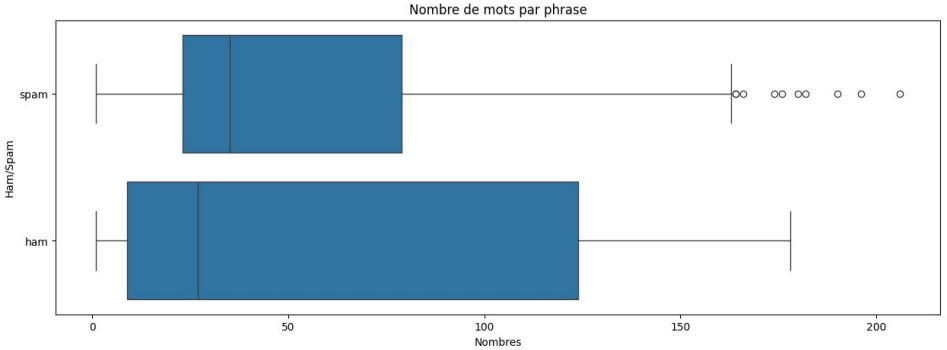
II

Visualisation des données

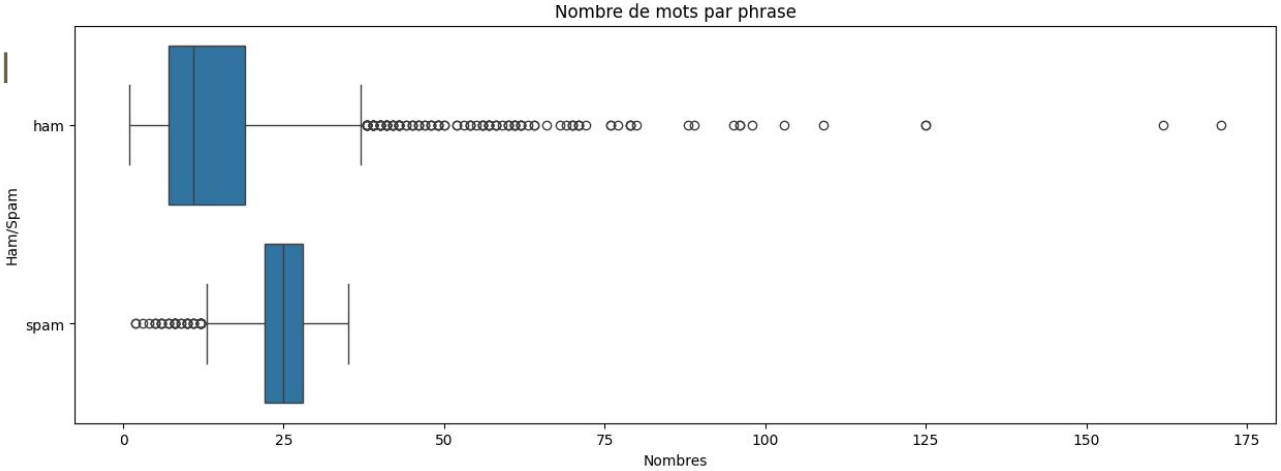
Dataset Niégerien



Dataset Télégram



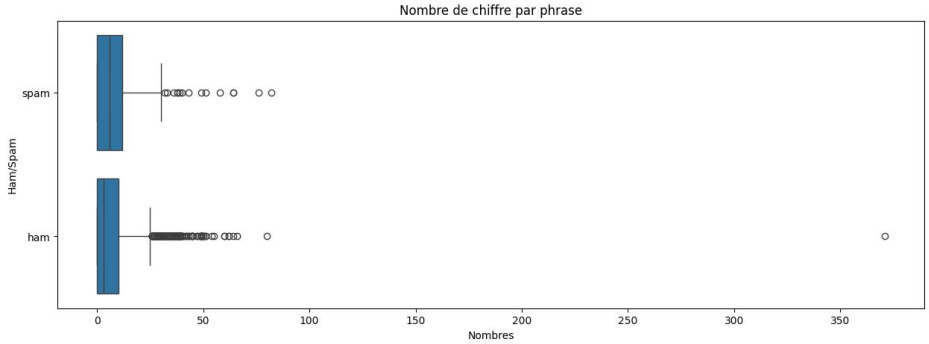
Dataset original



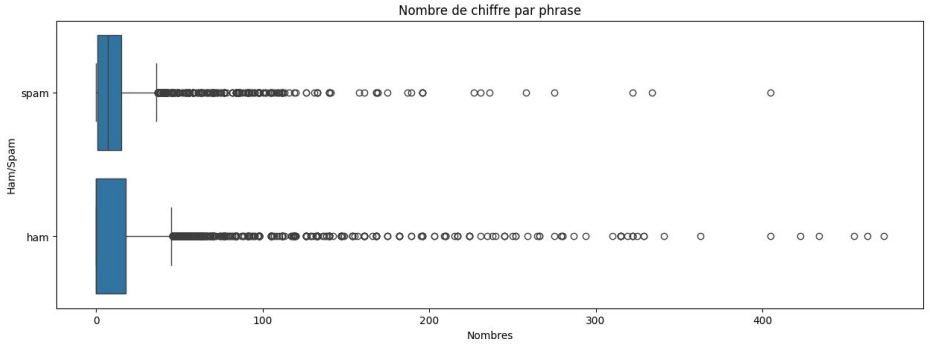
II

Visualisation des données

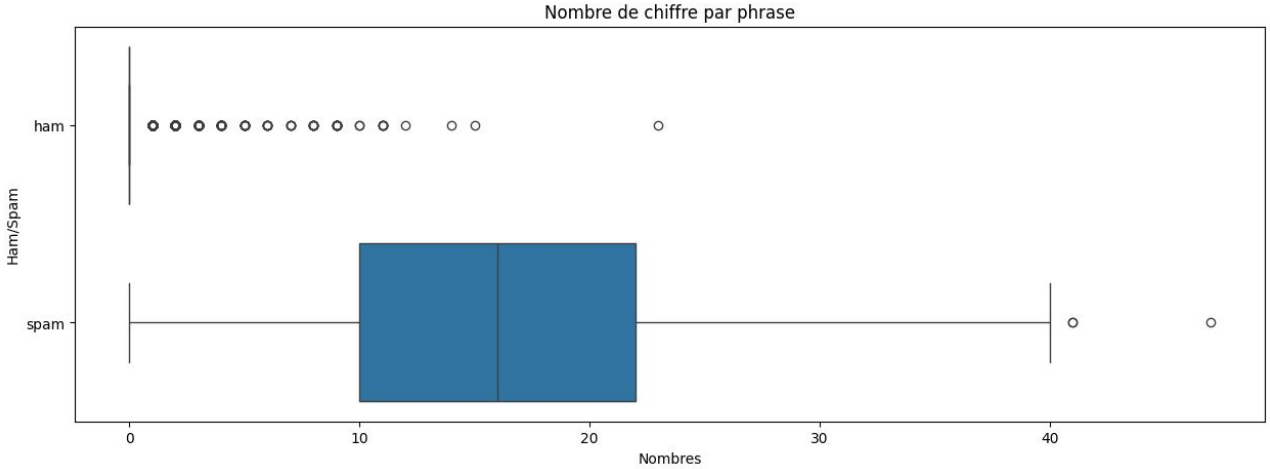
Dataset Niégerien



Dataset Télégram

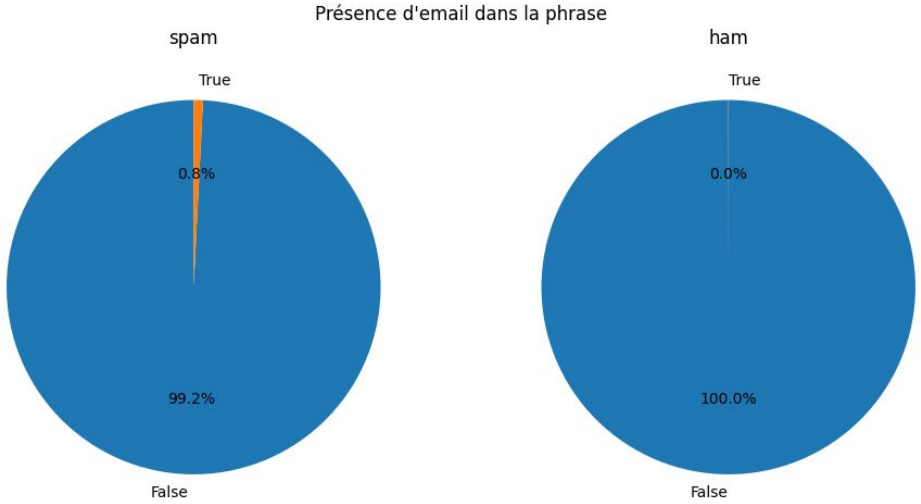


Dataset original

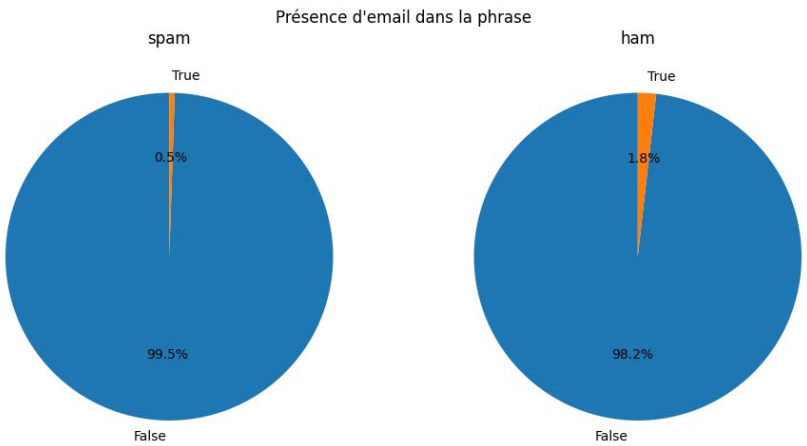


II

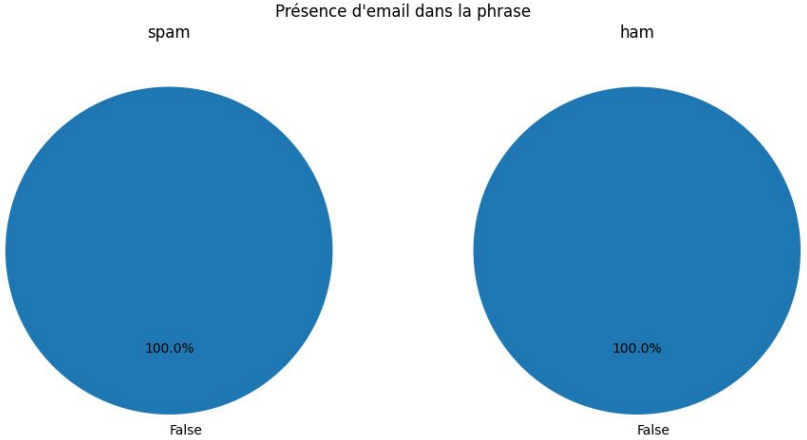
Visualisation des données



Dataset original



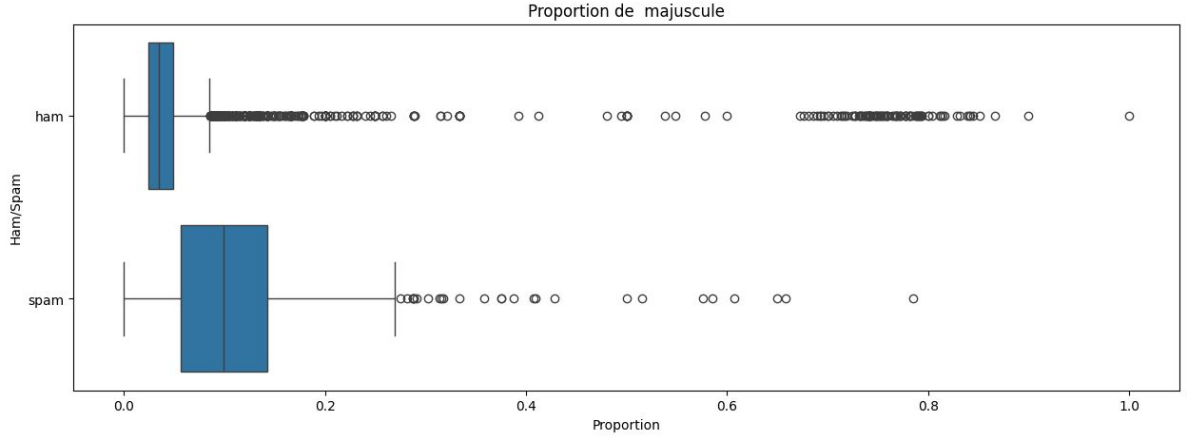
Dataset Nigérien

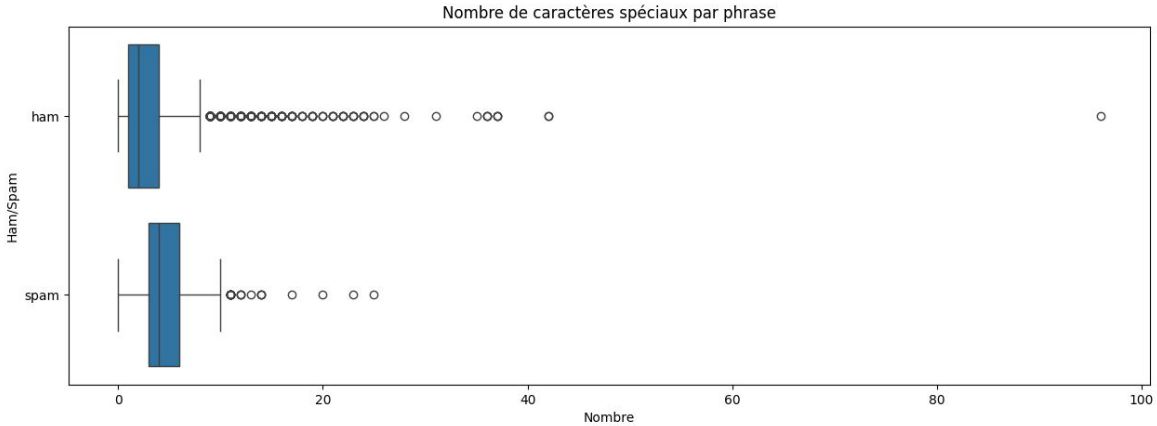


Dataset Télégram

II

Visualisation des données





III

Tests des modèles

Guidelines de test :

- Entraînement sur le même découpage des jeux de données
- Test sur les mêmes jeux de données non disponible en entraînement
- Test final sur des spams écrits mains

III

Gestion base de données

```
import Automatisation.DataSetManagement as DataSetManagement
import Automatisation.EvaluateModelsFeatures as EvaluateModelsFeatures

dataSet = {"first":"DataSetBrut/BD1.txt","nigerian":"DataSetBrut/DataSmsSpamNH.csv","telegram":"DataSetBrut/
telegram_spam_dataset.csv"}

dataManag = DataSetManagement.DataSetManage(dataSet,42)
```

✓ 0.4s

```
testBat = dataManag.GetAllTest()
Xtrain,ytrain = dataManag.GetCombinedTrain(["first","nigerian","telegram"])
```

- Système modulaire de gestion des base de données et séparation Train/Test

Résultat du test automatique

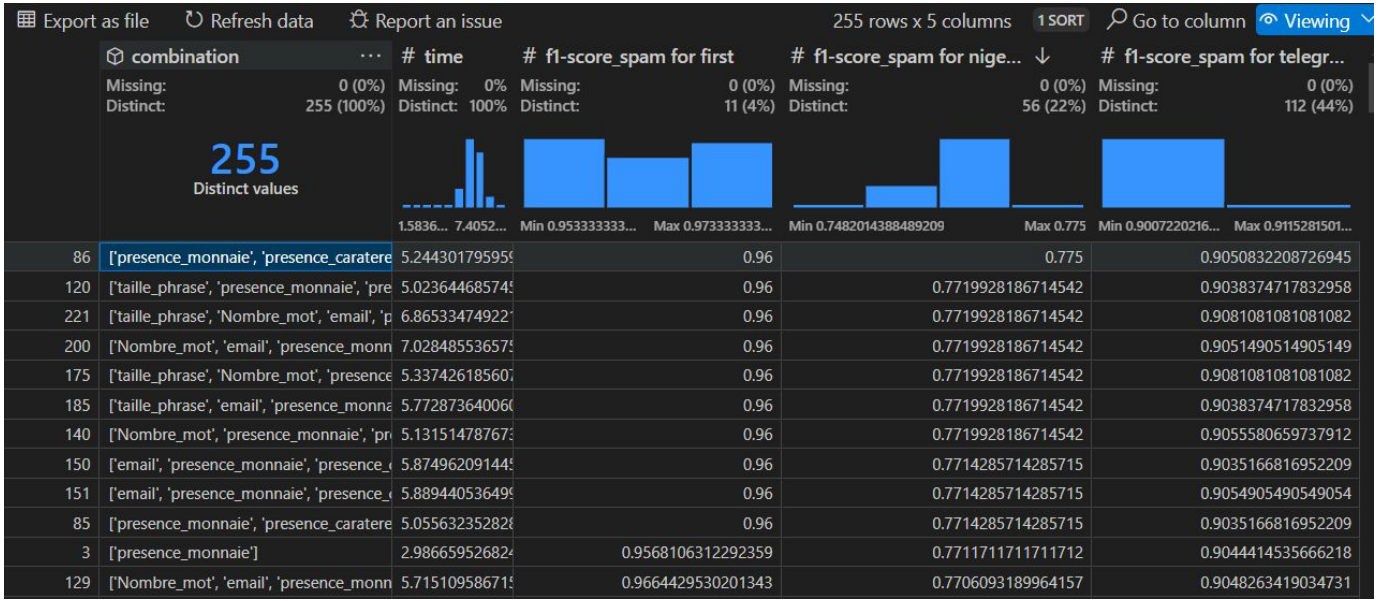
```
evalll = EvaluateModelsFeatures.calculatePrecisionOnBattery(X_train=Xtrain,y_train=ytrain,testBattery=testBat)
```

Export as file | Refresh data | Report an issue | 255 rows x 42 columns | Go to column | Viewing

	combination	# time	dictionnary	# precision_ham for first	# recall_ham for first	# f1-sco
	Missing: 0 (0%) Distinct: 255 (100%)	Missing: 0 (0%) Distinct: 255 (100%)	Missing: 0 (0%) Distinct: 112 (44%)	Missing: 0 (0%) Distinct: 11 (4%)	Missing: 0 (0%) Distinct: 5 (2%)	Missing: Distinct:
	255 Distinct values	Min 1.5836644172... Max 7.405297756...	112 Distinct values	Min 0.993775933... Max 0.996887966...	Min 0.991718426... Max 0.995859213...	Min 0.99274
0	['taille_phrase']	3.9081130027770996	{'ham': {'precision': 0.9591354272205	0.9937823834196892	0.9927536231884058	
1	['Nombre_mot']	3.0889079570770264	{'ham': {'precision': 0.9585439838220	0.9937823834196892	0.9927536231884058	
2	['email']	5.218014478683472	{'ham': {'precision': 0.9601216627238	0.9937759336099585	0.9917184265010351	
3	['presence_monnaie']	2.986659526824951	{'ham': {'precision': 0.9595141700404	0.9948078920041536	0.9917184265010351	
4	['presence telephone']	2.561432123184204	{'ham': {'precision': 0.9592318059299	0.9968814968814969	0.9927536231884058	
5	['presence_caratere_speci	1.5836644172668457	{'ham': {'precision': 0.9579266240323	0.9937823834196892	0.9927536231884058	
6	['proportion_majuscule']	4.762545585632324	{'ham': {'precision': 0.9591630104623	0.9937759336099585	0.9917184265010351	
7	['presence_lien']	4.855851411819458	{'ham': {'precision': 0.9604863221884	0.9937823834196892	0.9927536231884058	
8	['taille_phrase', 'Nombre_	4.84261679649353	{'ham': {'precision': 0.9620724686759	0.9937759336099585	0.9917184265010351	
9	['taille_phrase', 'email']	5.174443006515503	{'ham': {'precision': 0.9591354272205	0.9937823834196892	0.9927536231884058	
10	['taille_phrase', 'presence_	4.013068199157715	{'ham': {'precision': 0.9605662285136	0.994824016563147	0.994824016563147	
11	['taille_phrase', 'presence	4.587756872177124	{'ham': {'precision': 0.9588671611598	0.9968847352024922	0.9937888198757764	
12	['taille_phrase', 'presence	3.913673162460327	{'ham': {'precision': 0.9588532883642	0.9937823834196892	0.9927536231884058	

- Test long à effectuer chaque modèle 5 secondes
- Beaucoup d'information a résumé

Resumé



Export as file | Refresh data | Report an issue | 255 rows x 5 columns | 1 SORT | Go to column | Viewing

	combination	# time	# f1-score_spam for first	# f1-score_spam for nige...	# f1-score_spam for telegr...
	Missing: 0 (0%) Distinct: 255 (100%)	Missing: 0% Distinct: 100%	Missing: 0 (0%) Distinct: 11 (4%)	Missing: 0 (0%) Distinct: 56 (22%)	Missing: 0 (0%) Distinct: 112 (44%)
	255 Distinct values	1.5836... 7.4052...	Min 0.953333333... Max 0.973333333...	Min 0.7482014388489209 Max 0.775	Min 0.9007220216... Max 0.9115281501...
86	['presence_monnaie', 'presence_caratere	5.244301795959	0.96	0.775	0.9050832208726945
120	['taille_phrase', 'presence_monnaie', 'pre	5.023644685745	0.96	0.7719928186714542	0.9038374717832958
221	['taille_phrase', 'Nombre_mot', 'email', 'p	6.865334749221	0.96	0.7719928186714542	0.9081081081081082
200	['Nombre_mot', 'email', 'presence_monn	7.028485536575	0.96	0.7719928186714542	0.9051490514905149
175	['taille_phrase', 'Nombre_mot', 'presence	5.337426185607	0.96	0.7719928186714542	0.9081081081081082
185	['taille_phrase', 'email', 'presence_monna	5.772873640060	0.96	0.7719928186714542	0.9038374717832958
140	['Nombre_mot', 'presence_monnaie', 'pr	5.131514787673	0.96	0.7719928186714542	0.9055580659737912
150	['email', 'presence_monnaie', 'presence_	5.874962091445	0.96	0.7714285714285715	0.9035166816952209
151	['email', 'presence_monnaie', 'presence_	5.889440536499	0.96	0.7714285714285715	0.9054905490549054
85	['presence_monnaie', 'presence_caratere	5.055632352828	0.96	0.7714285714285715	0.9035166816952209
3	['presence_monnaie']	2.986659526824	0.9568106312292359	0.7711711711711712	0.9044414535666218
129	['Nombre_mot', 'email', 'presence_monn	5.715109586715	0.9664429530201343	0.7706093189964157	0.9048263419034731

- Score correct sur toutes les data
- Influence faible du choix des features

Choix du modèle

Modèle choisi : Linear SVC entraîné sur les trois bases des données

sur features : `['presence_monnaie', 'presence_caratere_speciaux', 'presence_lien']`

Résultat sur test manuel

```
modelPipe.predict(pd.DataFrame(["Join us today ! Flexible work without constainst. Earn between 50 and 3600 euros per
day payement sent daily. To know more add : https:/wam.me/nawak5546431645",
"Chronopost: your package 7d6595466533 is waiting clic here to comfirm the delivery of your package: https://
chronoquost.fr",
"I'm a nigerian prince i have 250 millions dollars ($250 000 000) in asset and gold and i need help to fight corruption
charge levied against me in England . You will be rewarded for your help please reply.",
"There was a hack on your amazon account clic on this link to resolve the issue https://www.amozon.com/reclamation . If
you don't you will be fined $250 dollars a day",
"Your energy provider want to talk to you about a reduction of your bill call this number 0648874598",
"URGENT: Your account has been compromised. Call 1-800-123-4567 immediately to secure your account!"],columns=["text"]))
```

22] ✓ 0.0s Python

```
array(['spam', 'spam', 'spam', 'spam', 'ham', 'spam'], dtype=object)
```

Résultats comparatifs sur tests manuels

Premier Model

résult
ham
ham
spam
ham
ham
ham

1/6

Naif Bayesien Main

result
spam
spam
ham
ham
ham
spam

3/6

Model SVC

result
spam
spam
spam
spam
ham
spam

5/6

Application

Démonstration !

Deploy

Spam Detector

To detect if the sms is a spam

Enter the sms here

Join us today ! Flexible work without constainst. Earn between 50 and 3600 euros per day payement sent daily. To know more add : https:/wam.me/nawak5546431645

Detect

The sms is a spam

Deploy

Spam Detector

To detect if the sms is a spam

Enter the sms here

Hello it's been a while ! What are you up to ? I'm coming to town this weekend , are you free for a beer?

Detect

The sms is not a spam

Piste de continuation

- Appliquer l’automatisation aux différents modèles
- Implémenter un système de barre de temps de chargement pour long test
- Faire un système de gestion de l’entraînement avec de nouvelles données. Gérez la pipeline d'insertion des données. (Concaténation sensible taille base de données)
- Tester le modèle avec des data actuel et concrète (Récupération de Data)
- Sérialiser un modèle pour le mettre en production
- Créer des variable environnement (localisation des dataset)
- Data Analyse : implémenter graphique en courbes

Est-ce que vous avez des questions ?

Data Extraction

```
class DataExtraction(BaseEstimator, TransformerMixin):

    def __init__(self, features_list):
        self.features_list = features_list
    def fit(self,X,y=None):
        return self
    def transform(self,X):
        retour = X
        for feature_name, feature_function in self.features_list.items():
            retour[feature_name] = X["text"].apply(feature_function)
        retour = retour.drop(columns = "text")
        return retour
```

List extraction features

```
features = {
    "taille_phrase":lambda x: len(x),
    "Nombre_mot":lambda x: len(x.split()),
    "email": lambda x: len(re.findall(r'[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}', x)) > 0,
    "presence_monnaie": lambda x: 1 if re.search(r'[\$\€\£]', x) else 0 ,
    "presence telephone": lambda x: 1 if re.search(r'\b\d{10,}\b', x) else 0,
    "presence_caratere_speciaux": lambda x: 1 if re.search(r'[!@#$%^&*(),.?":{}|<>]', x) else 0,
    "proportion_majuscule": lambda x: sum(1 for c in x if c.isupper()) / len(x) if len(x) > 0 else 0,
    "presence_lien": lambda x: 1 if re.search(r'\b(http|www)\S+', x) else 0
}
```