

Modèle	Précision
K nearest neighbors	24% avec le meilleur k =1
SVM	60% avec kernel='Linear' 56% avec kernel='rbf'
Régression logistique	précision max de 77,8% avec 5 sentiments, précision max de 89% avec 3 sentiments

K nearest neighbors :

```

import cst
import cleaning_text as cleaner
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from vectorizing_text import Vectorizer

def KNeighbors(k):
    X = cleaner.clean_text(cst.TRAIN)
    print(X)
    y = cleaner.read_column(cst.TRAIN, 5)
    vectorizer = Vectorizer(X)
    matrix = vectorizer.get_score_matrix()
    X_train, X_test, y_train, y_test = train_test_split(matrix, y)

    for k in range(1,30,5):
        model = KNeighborsClassifier(n_neighbors = k)
        model.fit(X_train,y_train)
        predictions = model.predict(X_test)
        scores = (accuracy_score(y_test,predictions))
        print(scores)

if __name__ == "__main__":
    KNeighbors(1)
  
```

```

haifa@ubuntu: ~/Documents/Fouille/covid-19_data_mining/src
File Edit View Search Terminal Help

still shocked by the number of toronto supermarket employees working without some
sort of mask we all know by now employees can be asymptomatic while spreading
coronavirus ", "i never that we'd be in a situation world that going to the s
supermarket picking up a package or my mail is like playing russian roulette with
my life coronavirus stay home social distancing", " you are definitely my man
i feel like this fall when we are out and about again we need to honor our hero
es from covid you know who they are our healthcare people grocery store owners a
nd anyone else putting their life on the li", 'airline pilots offering to stock
supermarket shelves in nz lockdown covid', 'response to complaint not provided
citing covid related delays yet prompt in rejecting policy before consumer tat i
s over way to go', "you know it's getting tough when is rationing toilet paper
coronavirus toilet paper martinsville help us out ", "is it wrong that the sme
ll of hand sanitizer is starting to turn me on coronavirus covid coronavirus",
"well new used rift s are going for $700 00 on amazon rn although the normal mar
ket price is usually $400 00 prices are really crazy right now for vr headsets s
ince hl alex was announced and it's only been worse with covid up to you whethe"
]
0.2468415937803693
0.20092322643343052
0.19233884029802398
0.1890184645286686
0.18658892128279883
0.1861839974084872
haifa@ubuntu:~/Documents/Fouille/covid-19_data_mining/src$
  
```

SVM :

Terminal window showing the execution of `svm.py` using an RBF kernel. The script imports necessary libraries, cleans the text, splits the data, and trains an SVM classifier. The output shows the confusion matrix and classification report.

```
import cst
from sklearn.svm import SVC
import cleaning_text as cleaner
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from vectorizing_text import Vectorizer

def svm_classify():
    X = cleaner.clean_text(cst.TRAIN)
    print(X)
    y = cleaner.read_column(cst.TRAIN, 5)
    vectorizer = Vectorizer(X)
    matrix = vectorizer.get_score_matrix()
    X_train, X_test, y_train, y_test = train_test_split(matrix, y)
    svclassifier = SVC(kernel='rbf')
    svclassifier.fit(X_train, y_train)
    y_predicted = svclassifier.predict(X_test)
    print(confusion_matrix(y_test, y_predicted))
    print(classification_report(y_test, y_predicted))

if __name__ == "__main__":
    svm_classify()
```

Confusion Matrix:

	Extremely Negative	Extremely Positive	Negative	Neutral	Positive
Extremely Negative	624	9	878	78	139
Extremely Positive	10	875	79	59	956
Negative	183	40	1744	319	657
Neutral	19	18	353	1407	501
Positive	31	192	530	359	2288

Classification Report:

	precision	recall	f1-score	support
Extremely Negative	0.72	0.36	0.48	1728
Extremely Positive	0.77	0.44	0.56	1979
Negative	0.49	0.59	0.53	2943
Neutral	0.63	0.61	0.62	2298
Positive	0.50	0.67	0.58	3400
accuracy			0.56	12348
macro avg	0.62	0.54	0.56	12348
weighted avg	0.60	0.56	0.56	12348

Terminal window showing the execution of `svm.py` using a Linear kernel. The script imports necessary libraries, cleans the text, splits the data, and trains an SVM classifier. The output shows the confusion matrix and classification report.

```
import cst
from sklearn.svm import SVC
import cleaning_text as cleaner
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from vectorizing_text import Vectorizer

def svm_classify():
    X = cleaner.clean_text(cst.TRAIN)
    print(X)
    y = cleaner.read_column(cst.TRAIN, 5)
    vectorizer = Vectorizer(X)
    matrix = vectorizer.get_score_matrix()
    X_train, X_test, y_train, y_test = train_test_split(matrix, y)
    svclassifier = SVC(kernel='linear')
    svclassifier.fit(X_train, y_train)
    y_predicted = svclassifier.predict(X_test)
    print(confusion_matrix(y_test, y_predicted))
    print(classification_report(y_test, y_predicted))

if __name__ == "__main__":
    svm_classify()
```

Confusion Matrix:

	Extremely Negative	Extremely Positive	Negative	Neutral	Positive
Extremely Negative	835	13	721	61	70
Extremely Positive	10	1105	55	51	751
Negative	310	42	1697	318	547
Neutral	25	27	364	1567	374
Positive	53	359	463	356	2174

Classification Report:

	precision	recall	f1-score	support
Extremely Negative	0.68	0.49	0.57	1700
Extremely Positive	0.71	0.56	0.63	1972
Negative	0.51	0.58	0.55	2914
Neutral	0.67	0.66	0.67	2357
Positive	0.56	0.64	0.59	3405
accuracy			0.60	12348
macro avg	0.63	0.59	0.60	12348
weighted avg	0.61	0.60	0.60	12348