```python
###CC1 Machine Learning AARAB_Ayoub


#Objectif : Construction des modèles de la classification pour prédire si un patient fume ou non en utilisant un Dataset d'assurance

#Partie 1 : Pretraitement des données
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

#1 charger les données
df = pd.read_csv("insuranceCC1.csv")
df = df.dropna(how='all')

# Pour la colonne 'region' (catégorielle), remplacer les valeurs manquantes par la valeur la plus fréquente
df['region'] = df['region'].fillna(df['region'].mode()[0])

# Pour la colonne 'charges' (numérique), remplacer les valeurs manquantes par la moyenne
df['charges'] = df['charges'].fillna(df['charges'].mean())


df.head()
# print(df.info())
```

	age	sex	bmi	children	smoker	region	charges
0	NaN	female	27.900	1.0	yes	Martil	16884.92400
1	18.0	male	33.770	1.0	no	Martil	1725.55230
2	28.0	male	NaN	3.0	no	Martil	4449.46200
3	33.0	male	22.705	NaN	no	Martil	21984.47061
4	32.0	male	28.880	0.0	no	Martil	3866.85520

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1337 non-null   float64
 1   sex       1338 non-null   object
 2   bmi       1337 non-null   float64
 3   children  1337 non-null   float64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   charges   1338 non-null   float64
dtypes: float64(4), object(3)
memory usage: 73.3+ KB
```

```
df.tail()
```

	age	sex	bmi	children	smoker	region	charges
1333	50.0	male	30.97	3.0	no	Martil	10600.548300
1334	18.0	female	31.92	0.0	no	Martil	2205.980800
1335	18.0	female	36.85	0.0	no	Martil	1629.833500
1336	21.0	female	25.80	0.0	no	Martil	2007.945000
1337	61.0	female	29.07	0.0	yes	Martil	13258.551706

```
# Gerer les valeurs manquantes
print(df.isnull().sum())
```

```
age         1
sex         0
bmi         1
children    1
smoker      0
region      0
charges     0
dtype: int64
```

```
df.shape
```

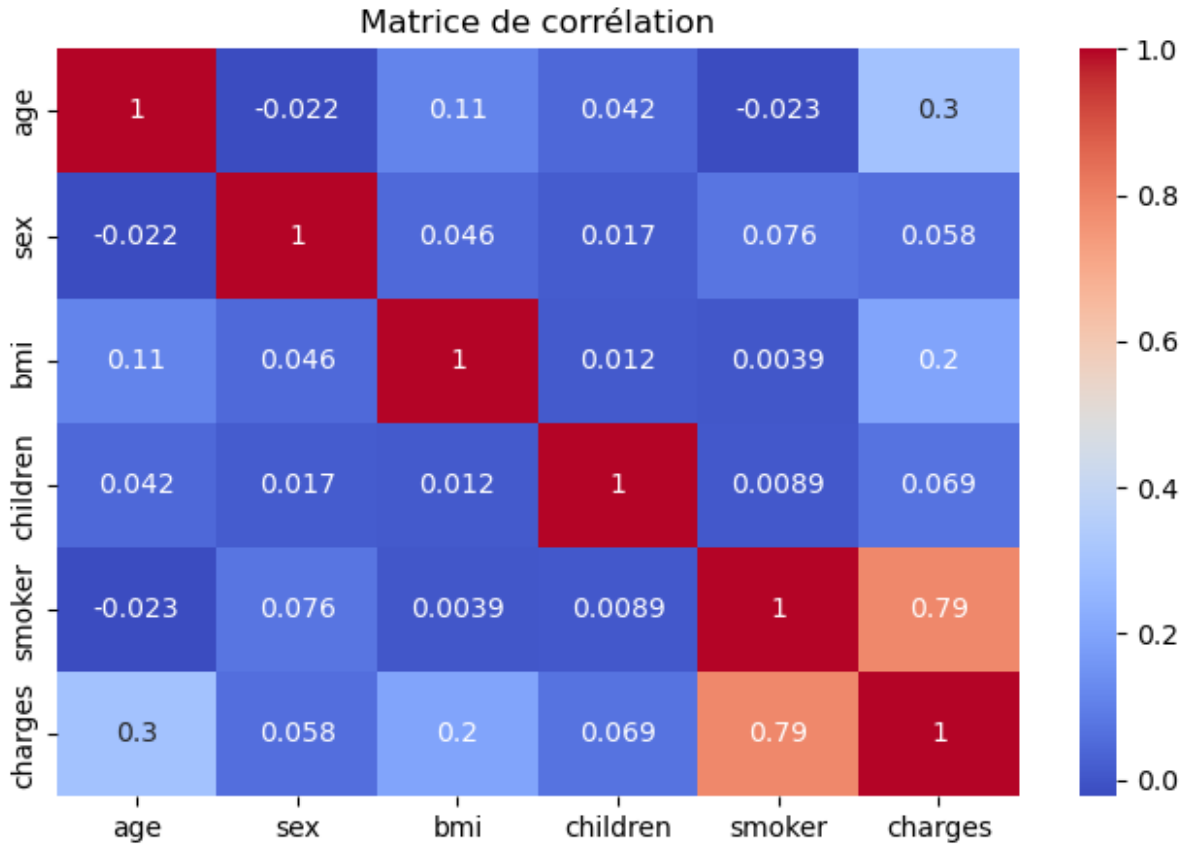
```
(1338, 7)
```

```
df.dtypes
```

```
age         float64
sex          object
bmi         float64
children    float64
smoker       object
region       object
charges     float64
dtype: object
```

```
from sklearn.impute import SimpleImputer

# Pour les colonnes numériques
imputer_mean = SimpleImputer(strategy='mean')
df['age'] = imputer_mean.fit_transform(df[['age']])
df['bmi'] = imputer_mean.fit_transform(df[['bmi']])

# Pour les colonnes discrètes/catégorielles
imputer_freq = SimpleImputer(strategy='most_frequent')
df['children'] = imputer_freq.fit_transform(df[['children']])

df['sex'] = df['sex'].map({'male': 1, 'female': 0})
df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})

# Supprimer la colonne 'region' qui est non numérique et inutile
# la colonne region contient uniquement la valeur "Martil" pour tous
```

```
les patients.
# Une colonne qui a la même valeur partout n'apporte aucune
information pour la classification, donc on la supprime.
df = df.drop(columns=['region'])

corr = df.corr()

plt.figure(figsize=(8,5))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title("Matrice de corrélation")
plt.show()
```



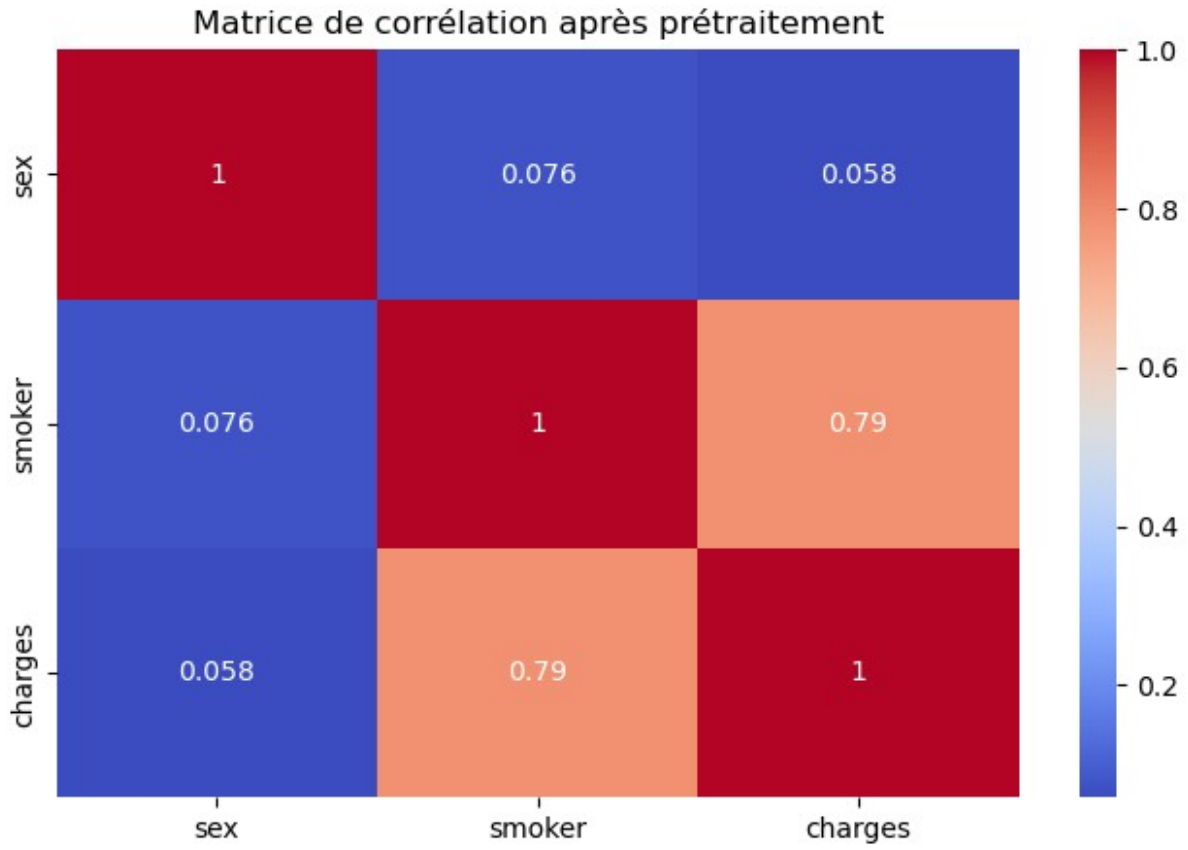

```
# Suppression des features les moins corrélées à 'smoker'
low_corr = corr['smoker'].abs().sort_values()

features_to_drop = low_corr[low_corr < 0.05].index.tolist()

features_to_drop = [f for f in features_to_drop if f != 'smoker']

df = df.drop(columns=features_to_drop)
```

```
print("Features supprimées :", features_to_drop)
```

```
Features supprimées : ['bmi', 'children', 'age']
```

```
# Afficher les données après le prétraitement
print(df.head())
# heatmap
plt.figure(figsize=(8,5))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Matrice de corrélation après prétraitement")
plt.show()
```

```
   sex  smoker      charges
0    0       1  16884.92400
1    1       0   1725.55230
2    1       0   4449.46200
3    1       0  21984.47061
4    1       0   3866.85520
```




```
# Partie 2 : Classification et comparaison des modèles

from sklearn.model_selection import train_test_split
```

```python
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report

# Séparer X et y
X = df.drop(columns=['smoker'])
y = df['smoker']


# Split train/test
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

#Dimensions du Dataset
print(f' Dimensions de X_train : {X_train.shape}')
print(f' Dimensions de y_train : {y_train.shape}')
print(f' Dimensions de X_test : {X_test.shape}')
print(f' Dimensions de y_test : {y_test.shape}')
```

```
 Dimensions de X_train : (936, 2)
 Dimensions de y_train : (936,)
 Dimensions de X_test : (402, 2)
 Dimensions de y_test : (402,)
```

```python
# 3. Normalisation (important pour SVM et régression logistique)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

X_train
```

```
array([[-1.02597835,  0.00552845],
       [ 0.97467943,  0.9296551 ],
       [-1.02597835,  1.1545818 ],
       ...,
       [ 0.97467943, -0.11835351],
       [-1.02597835,  2.70863924],
       [ 0.97467943, -0.26031267]], shape=(936, 2))
```

```python
X_test
```

```
array([[-1.02597835e+00, -3.52904507e-01],
       [-1.02597835e+00, -6.69069954e-01],
       [-1.02597835e+00,  1.32067045e+00],
       [ 9.74679434e-01, -3.35799393e-01],
       [ 9.74679434e-01,  1.68616143e+00],
       [ 9.74679434e-01, -7.29932630e-01],
       [-1.02597835e+00, -9.29985071e-01],
```

```
       [ 9.74679434e-01,  7.01610510e-02],
       [-1.02597835e+00, -7.96395726e-01],
       [ 9.74679434e-01, -2.56193546e-01],
       [ 9.74679434e-01,  4.04999855e-01],
       [ 9.74679434e-01, -5.04941538e-01],
       [-1.02597835e+00, -7.78632071e-01],
       [ 9.74679434e-01,  2.71175000e+00],
       [ 9.74679434e-01,  2.92036339e+00],
       [ 9.74679434e-01,  2.55060521e+00],
       [-1.02597835e+00, -2.94530938e-01],
       [-1.02597835e+00,  2.44864863e+00],
       [-1.02597835e+00, -4.24192248e-01],
       [-1.02597835e+00,  6.95710389e-01],
       [ 9.74679434e-01, -6.84955568e-01],
       [ 9.74679434e-01, -4.89659814e-01],
       [ 9.74679434e-01, -1.00119582e+00],
       [-1.02597835e+00, -8.77246720e-01],
       [-1.02597835e+00, -1.88282250e-01],
       [-1.02597835e+00, -2.01651520e-01],
       [-1.02597835e+00, -5.93477759e-02],
       [-1.02597835e+00,  4.50117462e-01],
       [-1.02597835e+00, -3.01564613e-01],
       [ 9.74679434e-01, -1.01151656e+00],
       [-1.02597835e+00,  2.03998467e-01],
       [-1.02597835e+00, -1.25672870e-01],
       [-1.02597835e+00, -9.37989629e-01],
       [-1.02597835e+00, -6.34230841e-01],
       [ 9.74679434e-01, -8.64918397e-01],
       [ 9.74679434e-01, -4.89088918e-01],
       [-1.02597835e+00, -8.90251133e-01],
       [-1.02597835e+00, -4.98240519e-01],
       [-1.02597835e+00,  8.70490419e-01],
       [ 9.74679434e-01,  2.09627703e+00],
       [-1.02597835e+00, -7.17399952e-01],
       [ 9.74679434e-01, -8.86489005e-01],
       [-1.02597835e+00, -1.39607843e-01],
       [ 9.74679434e-01, -1.02320078e-01],
       [ 9.74679434e-01, -7.00677158e-01],
       [ 9.74679434e-01, -8.50487057e-02],
       [-1.02597835e+00, -8.09032478e-01],
       [-1.02597835e+00, -7.41892092e-01],
       [ 9.74679434e-01,  2.37873652e+00],
       [ 9.74679434e-01, -7.35974421e-01],
       [-1.02597835e+00,  4.34204634e-02],
       [ 9.74679434e-01, -9.62892886e-01],
       [-1.02597835e+00,  1.25002139e+00],
       [ 9.74679434e-01, -9.63762142e-01],
       [ 9.74679434e-01, -2.28919386e-01],
       [ 9.74679434e-01,  9.90051969e-01],
```

```
       [ 9.74679434e-01, -8.03635209e-01],
       [ 9.74679434e-01,  2.09935186e+00],
       [-1.02597835e+00, -1.25216576e-01],
       [-1.02597835e+00, -2.31345997e-01],
       [-1.02597835e+00,  4.29031549e-02],
       [-1.02597835e+00, -7.11954790e-01],
       [ 9.74679434e-01,  1.20573818e+00],
       [-1.02597835e+00, -4.42059146e-01],
       [-1.02597835e+00,  8.00846339e-01],
       [-1.02597835e+00, -7.63230374e-01],
       [ 9.74679434e-01,  3.78773843e-01],
       [ 9.74679434e-01,  1.04459375e+00],
       [-1.02597835e+00, -8.07846336e-01],
       [ 9.74679434e-01, -9.65939614e-01],
       [ 9.74679434e-01, -6.02287474e-01],
       [-1.02597835e+00, -3.26775527e-01],
       [-1.02597835e+00, -4.20008856e-01],
       [ 9.74679434e-01, -5.80916627e-01],
       [ 9.74679434e-01, -5.13546946e-01],
       [-1.02597835e+00, -7.00756776e-01],
       [ 9.74679434e-01, -7.04239958e-01],
       [-1.02597835e+00, -1.57811520e-01],
       [-1.02597835e+00, -7.45381340e-01],
       [-1.02597835e+00, -3.52347936e-01],
       [ 9.74679434e-01, -9.90011895e-01],
       [ 9.74679434e-01,  1.21897445e+00],
       [-1.02597835e+00, -6.78997795e-01],
       [-1.02597835e+00,  2.07993920e+00],
       [ 9.74679434e-01,  2.57531235e+00],
       [ 9.74679434e-01,  2.29377110e+00],
       [ 9.74679434e-01, -7.05012228e-01],
       [ 9.74679434e-01, -2.28306841e-01],
       [-1.02597835e+00, -4.17762765e-01],
       [ 9.74679434e-01, -1.73482164e-01],
       [-1.02597835e+00,  1.54500792e-01],
       [-1.02597835e+00,  1.13102630e+00],
       [-1.02597835e+00,  8.30267142e-01],
       [ 9.74679434e-01, -6.88993672e-01],
       [ 9.74679434e-01,  2.06101019e+00],
       [ 9.74679434e-01, -5.36431783e-01],
       [ 9.74679434e-01,  5.25786120e-01],
       [-1.02597835e+00, -9.49607965e-01],
       [-1.02597835e+00,  4.46902214e-01],
       [-1.02597835e+00, -5.75605779e-01],
       [-1.02597835e+00, -7.30683248e-01],
       [-1.02597835e+00, -9.60926394e-01],
       [ 9.74679434e-01, -6.32929597e-01],
       [ 9.74679434e-01, -8.25908492e-02],
       [ 9.74679434e-01, -1.92355279e-02],
```

```
       [ 9.74679434e-01, -9.62222684e-01],
       [ 9.74679434e-01, -4.65681997e-01],
       [ 9.74679434e-01,  6.47866243e-01],
       [ 9.74679434e-01, -9.64297269e-01],
       [-1.02597835e+00,  8.17322805e-01],
       [ 9.74679434e-01,  8.03938543e-01],
       [ 9.74679434e-01, -8.63018151e-01],
       [-1.02597835e+00, -2.83694680e-02],
       [-1.02597835e+00,  2.01297016e+00],
       [-1.02597835e+00, -2.96395150e-01],
       [ 9.74679434e-01, -9.41337645e-01],
       [ 9.74679434e-01,  1.39750275e+00],
       [-1.02597835e+00,  7.53956422e-01],
       [-1.02597835e+00, -5.82927688e-01],
       [ 9.74679434e-01, -8.74431796e-01],
       [-1.02597835e+00, -6.42384136e-01],
       [ 9.74679434e-01, -5.04526541e-01],
       [-1.02597835e+00, -1.62535635e-01],
       [ 9.74679434e-01, -9.40770020e-01],
       [-1.02597835e+00, -7.14758449e-01],
       [-1.02597835e+00, -4.64215727e-01],
       [-1.02597835e+00, -4.97580663e-01],
       [ 9.74679434e-01, -3.36946226e-01],
       [-1.02597835e+00, -9.24338535e-02],
       [ 9.74679434e-01, -9.53146678e-01],
       [-1.02597835e+00, -7.76522180e-01],
       [ 9.74679434e-01, -6.14303563e-01],
       [-1.02597835e+00, -6.22396848e-01],
       [-1.02597835e+00, -4.48821787e-01],
       [-1.02597835e+00, -6.37846253e-01],
       [ 9.74679434e-01, -8.39451141e-02],
       [ 9.74679434e-01, -7.91725522e-02],
       [ 9.74679434e-01,  7.26286577e-01],
       [-1.02597835e+00,  2.44977388e+00],
       [-1.02597835e+00,  1.97726507e+00],
       [-1.02597835e+00, -6.69430345e-01],
       [-1.02597835e+00, -2.89120771e-01],
       [ 9.74679434e-01, -9.13007514e-01],
       [-1.02597835e+00,  1.66314348e+00],
       [ 9.74679434e-01, -9.28304976e-01],
       [ 9.74679434e-01,  8.22430290e-01],
       [ 9.74679434e-01, -6.69955403e-01],
       [-1.02597835e+00, -7.94024882e-01],
       [-1.02597835e+00, -1.59240909e-01],
       [ 9.74679434e-01, -7.12167072e-01],
       [-1.02597835e+00,  2.75395492e+00],
       [-1.02597835e+00, -8.82614749e-01],
       [ 9.74679434e-01, -1.01025202e+00],
       [ 9.74679434e-01,  2.00516719e+00],
```

```
       [-1.02597835e+00, -5.77982173e-01],
       [ 9.74679434e-01, -7.31374374e-01],
       [ 9.74679434e-01,  1.59465342e-02],
       [-1.02597835e+00, -3.98174334e-01],
       [-1.02597835e+00,  1.74319058e+00],
       [ 9.74679434e-01,  2.13069448e+00],
       [-1.02597835e+00,  2.02771922e-02],
       [-1.02597835e+00, -8.80901573e-01],
       [ 9.74679434e-01,  1.34495545e-01],
       [ 9.74679434e-01, -9.06930971e-01],
       [ 9.74679434e-01, -8.08068878e-01],
       [ 9.74679434e-01, -5.12774432e-01],
       [-1.02597835e+00,  2.93284258e+00],
       [-1.02597835e+00,  2.49902205e+00],
       [ 9.74679434e-01,  2.16635298e+00],
       [-1.02597835e+00, -8.87338905e-01],
       [-1.02597835e+00, -3.45700914e-01],
       [-1.02597835e+00, -5.89168554e-01],
       [ 9.74679434e-01, -6.28939246e-01],
       [ 9.74679434e-01, -7.54443076e-01],
       [-1.02597835e+00, -9.26923221e-01],
       [ 9.74679434e-01,  7.12611181e-01],
       [ 9.74679434e-01,  2.42788276e-01],
       [ 9.74679434e-01,  3.87817401e-02],
       [-1.02597835e+00,  1.60212625e+00],
       [-1.02597835e+00, -1.26666677e-01],
       [-1.02597835e+00,  1.33656455e+00],
       [-1.02597835e+00, -8.43137033e-01],
       [ 9.74679434e-01, -4.08067456e-01],
       [-1.02597835e+00, -6.90548372e-01],
       [ 9.74679434e-01, -6.74246922e-01],
       [-1.02597835e+00, -8.68941945e-01],
       [ 9.74679434e-01,  5.71943836e-01],
       [-1.02597835e+00, -8.11151805e-01],
       [-1.02597835e+00, -3.96340763e-01],
       [ 9.74679434e-01, -2.28896572e-01],
       [ 9.74679434e-01, -8.07563065e-01],
       [-1.02597835e+00, -4.99399454e-01],
       [-1.02597835e+00, -8.79157887e-01],
       [ 9.74679434e-01,  2.09642788e+00],
       [ 9.74679434e-01, -3.14969185e-02],
       [-1.02597835e+00, -3.37315471e-01],
       [-1.02597835e+00, -8.81043545e-01],
       [-1.02597835e+00, -1.04663986e-01],
       [-1.02597835e+00, -9.23298435e-01],
       [-1.02597835e+00, -3.66383827e-01],
       [-1.02597835e+00, -8.42406611e-01],
       [ 9.74679434e-01,  9.12334906e-01],
       [ 9.74679434e-01, -7.52270202e-01],
```

```
       [ 9.74679434e-01,  7.55201903e-01],
       [-1.02597835e+00,  3.07910563e-01],
       [-1.02597835e+00,  2.65939672e-01],
       [ 9.74679434e-01, -3.75057251e-01],
       [-1.02597835e+00, -7.49780389e-01],
       [ 9.74679434e-01, -4.59666712e-01],
       [-1.02597835e+00, -8.39718815e-01],
       [ 9.74679434e-01, -3.35006269e-02],
       [-1.02597835e+00,  7.30289835e-01],
       [-1.02597835e+00, -3.24962648e-01],
       [ 9.74679434e-01,  3.04443132e-01],
       [ 9.74679434e-01, -5.59791525e-01],
       [-1.02597835e+00, -7.62909398e-01],
       [-1.02597835e+00, -7.09807360e-01],
       [-1.02597835e+00,  5.28554680e-02],
       [-1.02597835e+00, -6.05628762e-02],
       [-1.02597835e+00, -6.71298714e-01],
       [-1.02597835e+00, -9.06216589e-01],
       [ 9.74679434e-01, -5.30415968e-01],
       [ 9.74679434e-01, -5.72849087e-01],
       [ 9.74679434e-01,  2.41478810e+00],
       [ 9.74679434e-01, -9.35113824e-01],
       [ 9.74679434e-01,  1.76240182e+00],
       [-1.02597835e+00, -9.61880541e-01],
       [ 9.74679434e-01, -9.69498656e-01],
       [ 9.74679434e-01, -3.29513027e-01],
       [-1.02597835e+00, -1.97245664e-01],
       [ 9.74679434e-01, -9.78203879e-01],
       [-1.02597835e+00, -3.07485211e-01],
       [-1.02597835e+00, -7.30493523e-01],
       [ 9.74679434e-01,  2.02354355e+00],
       [-1.02597835e+00, -2.78804969e-01],
       [ 9.74679434e-01, -4.33067377e-01],
       [ 9.74679434e-01, -7.84430297e-01],
       [ 9.74679434e-01, -6.10552873e-01],
       [-1.02597835e+00,  2.52527510e+00],
       [ 9.74679434e-01, -9.66598321e-01],
       [-1.02597835e+00, -1.30590708e-02],
       [ 9.74679434e-01,  2.58226419e+00],
       [-1.02597835e+00, -8.55182481e-01],
       [-1.02597835e+00, -8.24485728e-01],
       [ 9.74679434e-01, -9.70121150e-01],
       [ 9.74679434e-01, -8.81832827e-01],
       [-1.02597835e+00,  5.63662237e-01],
       [ 9.74679434e-01, -7.67956419e-01],
       [-1.02597835e+00, -6.36528497e-02],
       [ 9.74679434e-01, -9.71005748e-01],
       [-1.02597835e+00, -9.49998820e-01],
       [ 9.74679434e-01, -5.09891829e-01],
```

```
       [-1.02597835e+00, -8.43633405e-01],
       [-1.02597835e+00, -1.10235246e-01],
       [-1.02597835e+00, -8.80202632e-01],
       [-1.02597835e+00,  4.01604507e-01],
       [ 9.74679434e-01, -1.00502378e-01],
       [ 9.74679434e-01, -8.32793013e-01],
       [ 9.74679434e-01, -3.86499325e-01],
       [-1.02597835e+00, -5.99511992e-01],
       [-1.02597835e+00, -4.38534505e-01],
       [-1.02597835e+00,  4.37083163e-03],
       [ 9.74679434e-01,  4.03947829e-01],
       [ 9.74679434e-01,  2.78654740e+00],
       [-1.02597835e+00, -9.42229470e-02],
       [ 9.74679434e-01, -6.03324965e-01],
       [-1.02597835e+00,  4.16892281e+00],
       [-1.02597835e+00, -2.88591789e-01],
       [-1.02597835e+00, -3.44735891e-01],
       [-1.02597835e+00, -3.99251088e-01],
       [-1.02597835e+00,  1.13764914e+00],
       [-1.02597835e+00, -3.94110588e-01],
       [-1.02597835e+00, -7.33442756e-02],
       [ 9.74679434e-01,  3.57292767e-03],
       [-1.02597835e+00, -9.27245102e-01],
       [ 9.74679434e-01,  3.18972898e-02],
       [ 9.74679434e-01, -5.58418536e-01],
       [ 9.74679434e-01,  2.12423972e+00],
       [ 9.74679434e-01,  1.15897921e+00],
       [ 9.74679434e-01, -3.56794945e-01],
       [-1.02597835e+00, -3.61746265e-01],
       [-1.02597835e+00,  1.27307372e-01],
       [ 9.74679434e-01, -2.70044025e-01],
       [ 9.74679434e-01, -3.74391072e-01],
       [ 9.74679434e-01,  2.07198693e+00],
       [ 9.74679434e-01, -8.80033830e-01],
       [-1.02597835e+00, -2.88185989e-01],
       [ 9.74679434e-01,  2.80427251e+00],
       [ 9.74679434e-01, -7.54671841e-01],
       [ 9.74679434e-01, -8.95493296e-01],
       [-1.02597835e+00, -9.22654098e-01],
       [ 9.74679434e-01, -1.98218279e-01],
       [-1.02597835e+00,  1.35996397e-01],
       [ 9.74679434e-01, -2.17122184e-01],
       [ 9.74679434e-01, -2.95525220e-01],
       [ 9.74679434e-01, -2.43087193e-01],
       [ 9.74679434e-01, -3.35567133e-01],
       [-1.02597835e+00, -8.25649098e-01],
       [-1.02597835e+00, -9.26814011e-01],
       [ 9.74679434e-01,  2.04806689e+00],
       [ 9.74679434e-01,  1.83017650e+00],
```

```
       [ 9.74679434e-01, -5.79411834e-01],
       [-1.02597835e+00,  8.84716816e-01],
       [ 9.74679434e-01,  2.55445893e-01],
       [ 9.74679434e-01, -3.60836133e-02],
       [ 9.74679434e-01,  1.95080877e+00],
       [-1.02597835e+00, -9.01669046e-01],
       [ 9.74679434e-01,  2.77279627e+00],
       [ 9.74679434e-01, -6.81308961e-01],
       [-1.02597835e+00,  6.37517017e-02],
       [-1.02597835e+00, -5.74622715e-01],
       [ 9.74679434e-01, -9.62817014e-01],
       [ 9.74679434e-01,  2.03102292e-01],
       [-1.02597835e+00,  8.29773679e-03],
       [ 9.74679434e-01, -9.31164324e-01],
       [-1.02597835e+00, -1.03947049e-01],
       [ 9.74679434e-01,  6.13599902e-01],
       [-1.02597835e+00, -9.31526932e-02],
       [ 9.74679434e-01,  2.36424103e+00],
       [-1.02597835e+00, -5.94479460e-02],
       [ 9.74679434e-01,  6.50170173e-01],
       [-1.02597835e+00, -4.47231748e-01],
       [ 9.74679434e-01,  1.49517507e-01],
       [ 9.74679434e-01, -8.02456143e-01],
       [-1.02597835e+00, -4.24230184e-01],
       [ 9.74679434e-01, -4.41157115e-01],
       [ 9.74679434e-01,  4.61181509e-02],
       [ 9.74679434e-01, -2.12566126e-01],
       [-1.02597835e+00,  3.73546560e-01],
       [ 9.74679434e-01, -2.28325056e-01],
       [-1.02597835e+00, -1.19997379e-02],
       [ 9.74679434e-01, -1.17239575e-01],
       [ 9.74679434e-01,  8.23869347e-02],
       [ 9.74679434e-01,  1.58675620e+00],
       [ 9.74679434e-01, -6.33700962e-01],
       [ 9.74679434e-01, -9.10497749e-01],
       [-1.02597835e+00, -7.46127413e-01],
       [ 9.74679434e-01, -3.28401389e-01],
       [ 9.74679434e-01,  2.38590051e+00],
       [-1.02597835e+00, -3.75382349e-01],
       [-1.02597835e+00,  7.39370187e-02],
       [-1.02597835e+00, -5.15099195e-01],
       [-1.02597835e+00, -6.65762318e-01],
       [ 9.74679434e-01, -7.82046564e-01],
       [-1.02597835e+00, -9.22861596e-01],
       [-1.02597835e+00,  1.23872859e+00],
       [ 9.74679434e-01, -6.51512767e-01],
       [-1.02597835e+00, -9.70935591e-01],
       [-1.02597835e+00, -1.70603928e-01],
       [-1.02597835e+00, -4.41264745e-01],
```

```
       [ 9.74679434e-01, -1.34338091e-01],
       [-1.02597835e+00, -8.64134288e-01],
       [-1.02597835e+00, -2.89557432e-01],
       [-1.02597835e+00, -9.00543124e-01],
       [-1.02597835e+00,  2.20166158e+00],
       [ 9.74679434e-01,  1.89767540e+00],
       [-1.02597835e+00,  2.56696000e+00],
       [-1.02597835e+00, -8.42759729e-01],
       [ 9.74679434e-01, -4.64798503e-01],
       [-1.02597835e+00, -5.93549938e-01],
       [-1.02597835e+00, -9.49573478e-01],
       [-1.02597835e+00,  6.95463946e-01],
       [ 9.74679434e-01, -5.37877550e-01],
       [ 9.74679434e-01,  3.15705704e-01],
       [ 9.74679434e-01,  2.43133468e+00],
       [-1.02597835e+00, -6.52044590e-01],
       [-1.02597835e+00, -5.98765802e-02],
       [-1.02597835e+00, -6.09742990e-01],
       [-1.02597835e+00, -3.13874993e-01],
       [-1.02597835e+00, -2.47388099e-01],
       [-1.02597835e+00,  5.15457726e-01],
       [ 9.74679434e-01, -1.47712756e-01],
       [-1.02597835e+00, -5.79021235e-01],
       [ 9.74679434e-01, -7.71616557e-01],
       [-1.02597835e+00, -8.91285750e-01],
       [-1.02597835e+00, -1.40910488e-01],
       [ 9.74679434e-01,  6.38466363e-01],
       [ 9.74679434e-01, -2.11282052e-01],
       [ 9.74679434e-01, -3.00846005e-01],
       [-1.02597835e+00, -8.40342550e-01],
       [ 9.74679434e-01, -8.31365328e-01],
       [ 9.74679434e-01, -6.11380567e-01],
       [-1.02597835e+00, -4.21219358e-01],
       [-1.02597835e+00, -3.26451744e-01],
       [ 9.74679434e-01,  8.77083547e-01],
       [-1.02597835e+00,  2.85870577e+00],
       [-1.02597835e+00, -1.25687901e-01],
       [-1.02597835e+00, -4.91339222e-01],
       [ 9.74679434e-01, -3.40834198e-02],
       [-1.02597835e+00, -1.92576622e-01],
       [ 9.74679434e-01, -6.02061405e-01],
       [-1.02597835e+00, -7.75123148e-01],
       [-1.02597835e+00, -4.81748186e-01],
       [ 9.74679434e-01, -8.16930443e-01],
       [ 9.74679434e-01, -4.33820348e-01],
       [-1.02597835e+00,  3.52650672e-01],
       [-1.02597835e+00, -1.26124740e-01],
       [-1.02597835e+00, -3.15967000e-01],
       [ 9.74679434e-01, -7.41223932e-01],
       [-1.02597835e+00, -9.23080016e-01],
```

```
       [ 9.74679434e-01, -1.65315868e-01],
       [ 9.74679434e-01, -9.42602177e-01]])

# Logistic Regression
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
y_pred_logreg = logreg.predict(X_test)
print("Logistic Regression - Accuracy:", accuracy_score(y_test,
y_pred_logreg))
print(classification_report(y_test, y_pred_logreg))

Logistic Regression - Accuracy: 0.9129353233830846
              precision    recall  f1-score   support

           0       0.93      0.96      0.95       323
           1       0.82      0.71      0.76        79

    accuracy                           0.91       402
   macro avg       0.88      0.84      0.85       402
weighted avg       0.91      0.91      0.91       402


# Random Forest
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest - Accuracy:", accuracy_score(y_test, y_pred_rf))
print(classification_report(y_test, y_pred_rf))

Random Forest - Accuracy: 0.9228855721393034
              precision    recall  f1-score   support

           0       0.95      0.96      0.95       323
           1       0.82      0.77      0.80        79

    accuracy                           0.92       402
   macro avg       0.88      0.87      0.87       402
weighted avg       0.92      0.92      0.92       402


# SVM
from sklearn.svm import SVC
svm = SVC()
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
print("SVM - Accuracy:", accuracy_score(y_test, y_pred_svm))
print(classification_report(y_test, y_pred_svm))
```

```
SVM - Accuracy: 0.9303482587064676
              precision    recall  f1-score   support

           0       0.97      0.94      0.96       323
           1       0.79      0.87      0.83        79

    accuracy                           0.93       402
   macro avg       0.88      0.91      0.89       402
weighted avg       0.93      0.93      0.93       402
```

```
# K-Nearest Neighbors
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
y_pred_knn = knn.predict(X_test)
print("KNN - Accuracy:", accuracy_score(y_test, y_pred_knn))
print(classification_report(y_test, y_pred_knn))
```

```
KNN - Accuracy: 0.9353233830845771
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       323
           1       0.82      0.86      0.84        79

    accuracy                           0.94       402
   macro avg       0.89      0.91      0.90       402
weighted avg       0.94      0.94      0.94       402
```

```
# Decision Tree
from sklearn.tree import DecisionTreeClassifier
dt = DecisionTreeClassifier()
dt.fit(X_train, y_train)
y_pred_dt = dt.predict(X_test)
print("Decision Tree - Accuracy:", accuracy_score(y_test, y_pred_dt))
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree - Accuracy: 0.917910447761194
              precision    recall  f1-score   support

           0       0.94      0.95      0.95       323
           1       0.80      0.77      0.79        79

    accuracy                           0.92       402
   macro avg       0.87      0.86      0.87       402
weighted avg       0.92      0.92      0.92       402
```

```
# Gradient Boosting
```

```python
from sklearn.ensemble import GradientBoostingClassifier
gb = GradientBoostingClassifier()
gb.fit(X_train, y_train)
y_pred_gb = gb.predict(X_test)
print("Gradient Boosting - Accuracy:", accuracy_score(y_test,
y_pred_gb))
print(classification_report(y_test, y_pred_gb))
```

```
Gradient Boosting - Accuracy: 0.9328358208955224
              precision    recall  f1-score   support

           0       0.96      0.95      0.96       323
           1       0.82      0.85      0.83        79

    accuracy                           0.93       402
   macro avg       0.89      0.90      0.90       402
weighted avg       0.93      0.93      0.93       402
```