

# PROGRAMME

## En utilisant un centrage réduction et ACP sur fichier à tester

```
# Chargement du fichier à Evaluer et Préformattage (acp) avant application Modèle
ex = pd.read_csv("example.csv")
ex1 = ex.drop(columns = 'id').values
# Préparation Données pour modele Regression Logistique
x = df.copy()
x = df.drop(columns = 'is_genuine').values
y = df['is_genuine'].values
y.astype(int)

# Centrage et Réduction
std_scale = preprocessing.StandardScaler().fit(x)
x_scaled = std_scale.transform(x)
ex_scaled = std_scale.transform(ex1)

# Calcul des composantes principales
n_comp = 4 # choix du nombre de composantes à calculer
pca = decomposition.PCA(n_components=n_comp)
pca.fit(x_scaled)
x_projected = pca.fit_transform(x_scaled)
ex_projected = pca.fit_transform(ex_scaled)

# Création du Modèle de regression Logistique sur notre Dataset
logreg = LogisticRegression(solver='lbfgs')
logreg.fit(x_projected, y)

# Traitement Prédctions sur fichier Test (exemple) et sortie dans un Dataframe
e = ex.count()
if e["id"] > 0 :
    y_pred = logreg.predict(ex_projected)
    ex["predict"] = y_pred
    tx_conf = logreg.predict_proba(ex_projected)
    i = tx_conf[:,0]
    ex["Probabilité d'authenticité(%)"] = np.round((i*100), 2)
    ex = ex.replace({'predict' : 1}, "Le billet est Faux")
    ex = ex.replace({'predict' : 0}, "Le billet est Vrai")
else :
    print("Fichier vide")

print("Taille Base Apprentissage = ", tx_split, "(%)")
print("-----")
print("Base Apprentissage >>> ", "Xn = ", logreg.coef_, " Y = ", logreg.intercept_)
print("-----")
print("Nombre de billets : ",e["id"])
ex.head()
```

Taille Base Apprentissage = 80 (%)

-----									
Base Apprentissage >>>	Xn =	[-2.6685	1.6438	0.0206	1.0892]	Y =	[1.0423]	-----	
Nombre de billets : 5									
	diagonal	height_left	height_right	margin_low	margin_up	length	id	predict	Probabilité d'authenticité(%)
0	171.76	104.01	103.54	5.21	3.30	111.42	A_1	False	97.36
1	171.87	104.17	104.13	6.00	3.31	112.09	A_2	False	51.02
2	172.00	104.58	104.29	4.99	3.39	111.57	A_3	True	2.49
3	172.49	104.55	104.34	4.44	3.03	113.20	A_4	True	0.00
4	171.65	103.63	103.56	3.77	3.16	113.33	A_5	False	99.12

## En utilisant juste un centrage réduction sur fichier à tester

```
# Préparation Données pour modele Regression Logistique
x = df.copy()
x = df.drop(columns = 'is_genuine').values
y = df['is_genuine'].values
y.astype(int)
tx_split = 80

# Centrage et Réduction
std_scale = preprocessing.StandardScaler().fit(x)
X_scaled = std_scale.transform(x)
# Regression Logistique
logreg = LogisticRegression(solver='lbfgs')
logreg.fit(X_scaled, y)

# Chargement du fichier à Evaluer et Préformattage (acp) avant application Modèle
ex = pd.read_csv("example.csv")
ex1 = ex.drop(columns = 'id').values
# Centrage et Réduction
std_scale = preprocessing.StandardScaler().fit(ex1)
ex1_scaled = std_scale.transform(ex1)

# Traitement Prédctions et sortie dans un Dataframe
e = ex.count()
if e["id"] > 0 :
    y_pred = logreg.predict(ex1_scaled)
    ex["predict"] = y_pred
    tx_conf = logreg.predict_proba(ex1_scaled)
    i = tx_conf[:,0]
    ex["Probabilité d'authenticité(%)"] = np.round((i*100), 2)
    ex = ex.replace({'predict' : 1}, "Le billet est Faux")
    ex = ex.replace({'predict' : 0}, "Le billet est Vrai")
else :
    print("Fichier vide")

print("Taille Base Apprentissage = ", tx_split, "(%)")
print("-----")
print("Base Apprentissage >>> ", "Xn = ", logreg.coef_, " Y = ", logreg.intercept_)
print("-----")
print("Nombre de billets : ",e["id"])
ex.head()
```

Taille Base Apprentissage = 80 (%)

-----									
Base Apprentissage >>>	Xn =	[[ 0.1765	-0.1077	-0.4605	-2.3992	-1.4719	1.8039]	Y =	[0.9905]
-----									
Nombre de billets : 5									
	diagonal	height_left	height_right	margin_low	margin_up	length	id	predict	Probabilité d'authenticité(%)
0	171.76	104.01	103.54	5.21	3.30	111.42	A_1	False	90.89
1	171.87	104.17	104.13	6.00	3.31	112.09	A_2	False	98.52
2	172.00	104.58	104.29	4.99	3.39	111.57	A_3	False	96.47
3	172.49	104.55	104.34	4.44	3.03	113.20	A_4	True	0.15
4	171.65	103.63	103.56	3.77	3.16	113.33	A_5	True	0.03