

detection_faux_billets

```
options(repos = c(CRAN = "https://cran.rstudio.com/"))
```

```
new_data <- read.csv("data_raw/billets_production.csv", sep = ",")
```

```
head(new_data)
```

	diagonal	height_left	height_right	margin_low	margin_up	length	id
1	171.76	104.01	103.54	5.21	3.30	111.42	A_1
2	171.87	104.17	104.13	6.00	3.31	112.09	A_2
3	172.00	104.58	104.29	4.99	3.39	111.57	A_3
4	172.49	104.55	104.34	4.44	3.03	113.20	A_4
5	171.65	103.63	103.56	3.77	3.16	113.33	A_5

```
# Extraire la colonne 'id' pour référence  
ids <- new_data$id
```

```
# Supprimer la colonne 'id' avant de faire des prédictions  
new_data <- new_data[, -which(names(new_data) == "id")]
```

```
# Normaliser les données si nécessaire (utiliser les statistiques de l'entraînement)  
# Par exemple, avec scale (supposons que c'est ce que vous aviez fait)  
new_data_scaled <- scale(new_data)
```

```
# Charger le modèle ajusté et le seuil optimal  
reg_log_model <- readRDS("modele_regression_logistique.rds")  
best_threshold <- readRDS("seuil_optimal.rds")
```

```
# Faire des prédictions de probabilités sur de nouvelles données  
predicted_prob_new <- predict(reg_log_model, new_data, type = "response")
```

```
# Appliquer le seuil optimal pour classer les probabilités
final_predictions <- ifelse(predicted_prob_new > best_threshold, "True", "False")
```

```
# Afficher les prédictions
print(final_predictions)
```

```
      1      2      3      4      5
"False" "False" "False" "True" "True"
```

```
table(final_predictions)
```

```
final_predictions
False True
      3   2
```

```
# Associer les identifiants de billets avec les prédictions
vertict <- data.frame(id = ids, is_genuine = final_predictions)
```

```
print(vertict)
```

```
   id is_genuine
1 A_1      False
2 A_2      False
3 A_3      False
4 A_4       True
5 A_5       True
```

ou

```
# Utilisez le nouvel environnement Conda
library(reticulate)
```

```
# Utilisez le nouvel environnement Conda
use_condaenv("tf-env", required = TRUE)
```

```
library(keras)
```

```
# Charger le modèle sauvegardé (réseau neuronal)
model <- load_model_hdf5("model_faux_billets.h5")
```

```
# Faire des prédictions sur le nouveau jeu de données
predictions <- model %>% predict(new_data_scaled)
```

1/1 - 1s - 844ms/epoch - 844ms/step

```
# Convertir les probabilités en classes (0 ou 1)
predicted_class <- ifelse(predictions > 0.5, "True", "False")

# Afficher les résultats
print(predicted_class)
```

```
      [,1]
[1,] "False"
[2,] "False"
[3,] "False"
[4,] "True"
[5,] "True"
```

```
# Associer les identifiants de billets avec les prédictions
results <- data.frame(id = ids, is_genuine = predicted_class)

# Afficher ou sauvegarder les résultats dans un fichier CSV
write.csv(results, "prediction_results.csv", row.names = FALSE)

print(results)
```

```
      id is_genuine
1 A_1      False
2 A_2      False
3 A_3      False
4 A_4       True
5 A_5       True
```