Using the RANDOM FOREST ALGORITHM on the PhiUSIIL Phishing URL Dataset

Presented By :

Abdelatif Mekri
Nahla Yasmine Mihoubi
Halima Nfidsa

Contenu de la presentation

ALGORITHME À IMPLÉMENTER

Principe de l'algorithme.
Avantages de l'algorithme.
Domaines d'applications.

LE DATASET

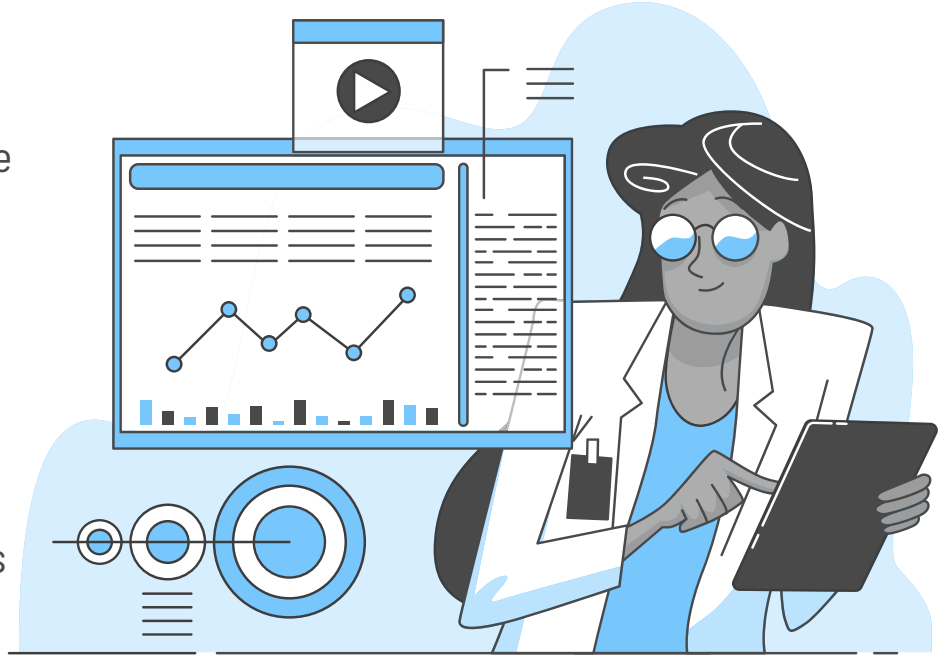
aperçu du Dataset utilisé et explication des raisons de ce choix.

IMPLÉMENTATION

Les différentes méthodes implémenter et leurs utilisations

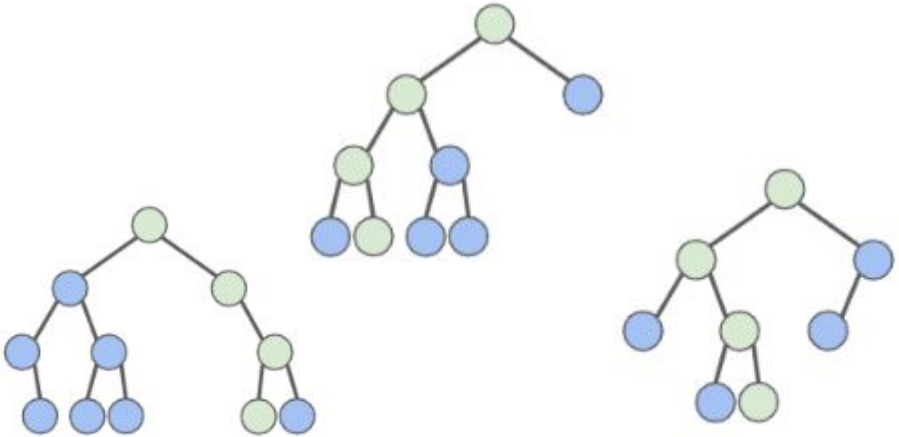
INTERPRÉTATION

Interpretations des resultats obtenues

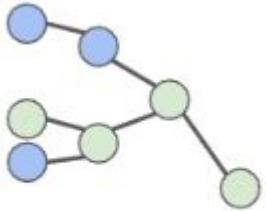


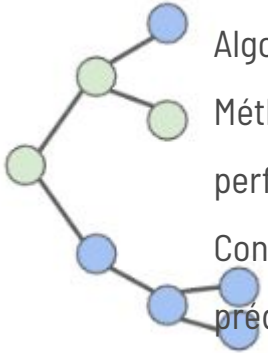
01

ALGORITHME À IMPLÉMENTER



Random Forest





Algorithme d'apprentissage supervisé polyvalent, utilisé pour la classification et la régression.

Méthodes d'ensemble combinent les prédictions de plusieurs modèles de base pour améliorer la performance prédictive.

Construit un ensemble de modèles d'arbres de décision lors de l'entraînement et effectue des prédictions en agrégeant les prédictions de ces arbres individuels.

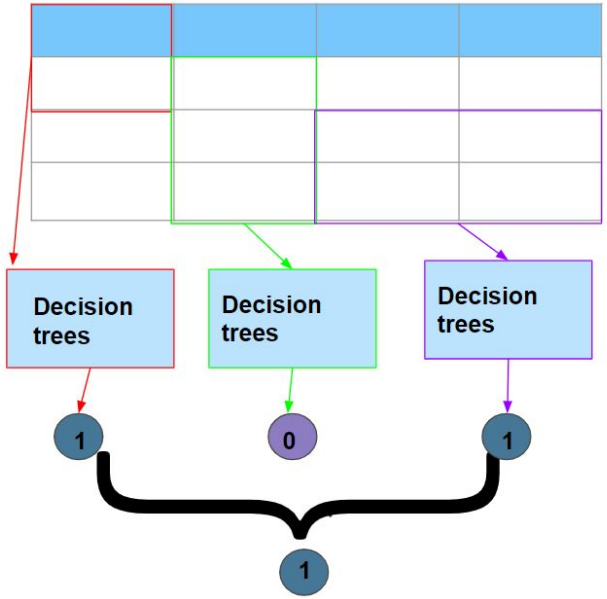
PRINCIPE DE L'ALGORITHME

01

Sélection aléatoire d'un échantillon de données.

02

Construction d'un arbre de décision basé sur le sous-échantillon sélectionné.



03

Répétition des étapes 1 et 2 pour former plusieurs arbres

04

Agrégation des prédictions des arbres pour obtenir une prédiction finale plus robuste et précise

AVANTAGES DE L'ALGORITHME

01 Données

Grande dimensionnalité et est moins sensible à la malédiction de la dimensionnalité.

02 Valeurs manquantes

Maintient une précision même lorsque qu'une grande proportion des données est manquante.

03 L'importance des caractéristiques

Permettant la sélection de caractéristiques et l'interprétation des résultats.

04 Surajustement

S'ajuste automatiquement grâce à l'effet de moyennage des multiples arbres.

DOMAINES D'APPLICATIONS

Prédiction de la santé des patients

Prédire les maladies ou les diagnostics en fonction des données

Détection de spam d'e-mails

Identifier les e-mails frauduleux en fonction de mots clés, les adresses IP, etc.

Systèmes de recommandation

Produits ou du contenu en fonction du comportement passé des utilisateurs

Classification de documents

Classer automatiquement les documents en catégories basées sur leur contenu.

Analyse de la clientèle

Segmenter les clients en groupes homogènes pour cibler les campagnes marketing, etc.

02

DATASET

DATASET

PhiUSIIL Phishing URL

Ensemble de données conséquent comprenant URL légitimes et URL de phishing.

La plupart des URL analysées lors de la construction de l'ensemble de données sont parmi les plus récentes.

Les caractéristiques sont extraites du code source de la page web et de l'URL.

134 850

URL légitimes

100 945

URL de phishing

56

Variables

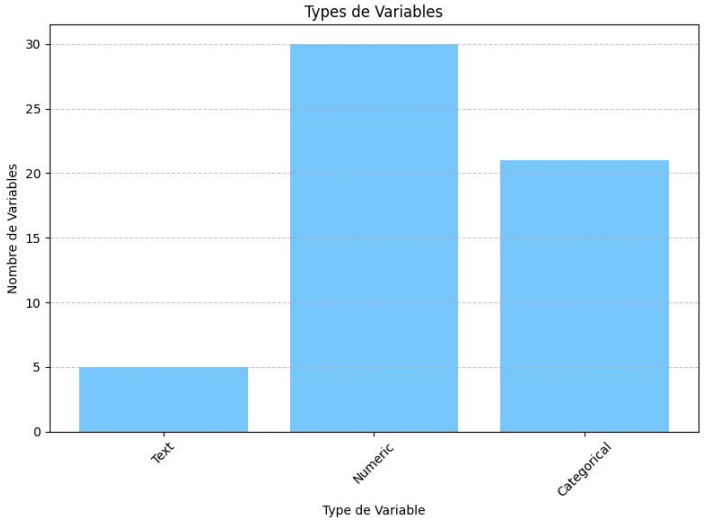
Dataset statistics

Overview

Dataset statistics

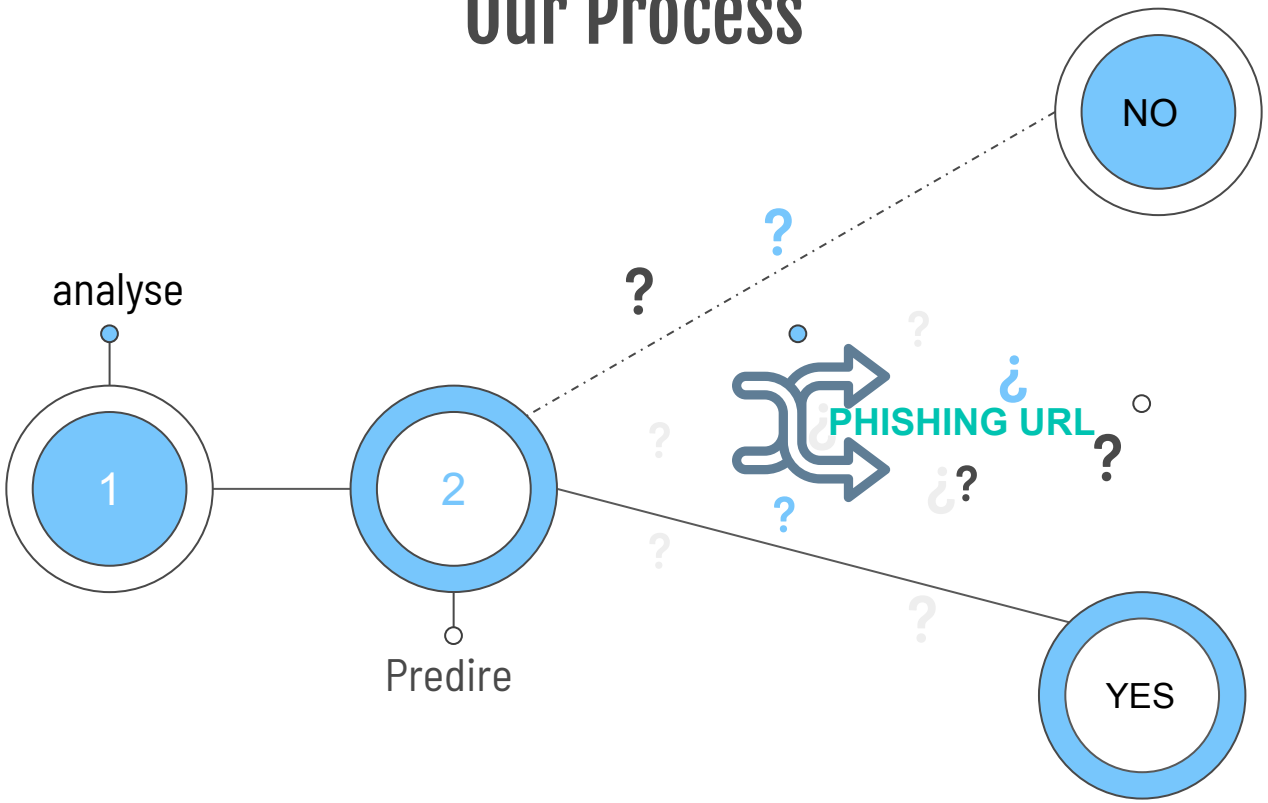
Number of variables	56
Number of observations	235795
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	175.2 MiB
Average record size in memory	779.0 B

Distributions de types des Variables



On voit que la majorité des variables sont numerique

Our Process



3

IMPLEMENTATION

IMPLEMENTATION

Installations
Imports des libraries
reading the files

ANALYSE DE DONNÉES ET PRÉTRAITEMENTS

Analyse univariate
Correlations des variables
Pretraitement

IMPLEMENTATION DU MODELE RANDOM FOREST

Data split
Initialisation des hyperparamètres de l’algorithme

Installing packages

```
[ ] pip install scikit-learn
    pip install pandas
    pip install numpy
    pip install matplotlib
    pip install seaborn
```

Importing Packages

```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix ,roc_curve, auc
    from sklearn.tree import plot_tree
    from sklearn.datasets import load_iris
    import pandas as pd
    import numpy as np
    from ydata_profiling import ProfileReport
    import matplotlib.pyplot as plt
    import seaborn as sns
```

CLEANING DATA

Reading csv

```
df= pd.read_csv("PhiUSIIL_Phishing_URL_Dataset.csv")
```

ANALYSE DES DONNEES

```python
# Generate profile report
profile = ProfileReport(df, title="Pandas Profiling Report", explorative=True)
profile.to_file(output_file="REPORT.html")
#Display the report as widgets
#profile.to_widgets()

print('Profiling Report found in "Pandas Profiling Report.html" ')
```

```
Profiling Report found in "Pandas Profiling Report.html"
```

Overview

Overview | Alerts 83 | Reproduction

Dataset statistics

Number of variables	56
Number of observations	235795
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	175.2 MiB
Average record size in memory	779.0 B

Variable types

Text	4
URL	1
Numeric	30
Categorical	21

Variables

Select Columns

FILENAME

Text

UNIQUE

Distinct	235795

PRÉTRAITEMENT DES DONNÉES

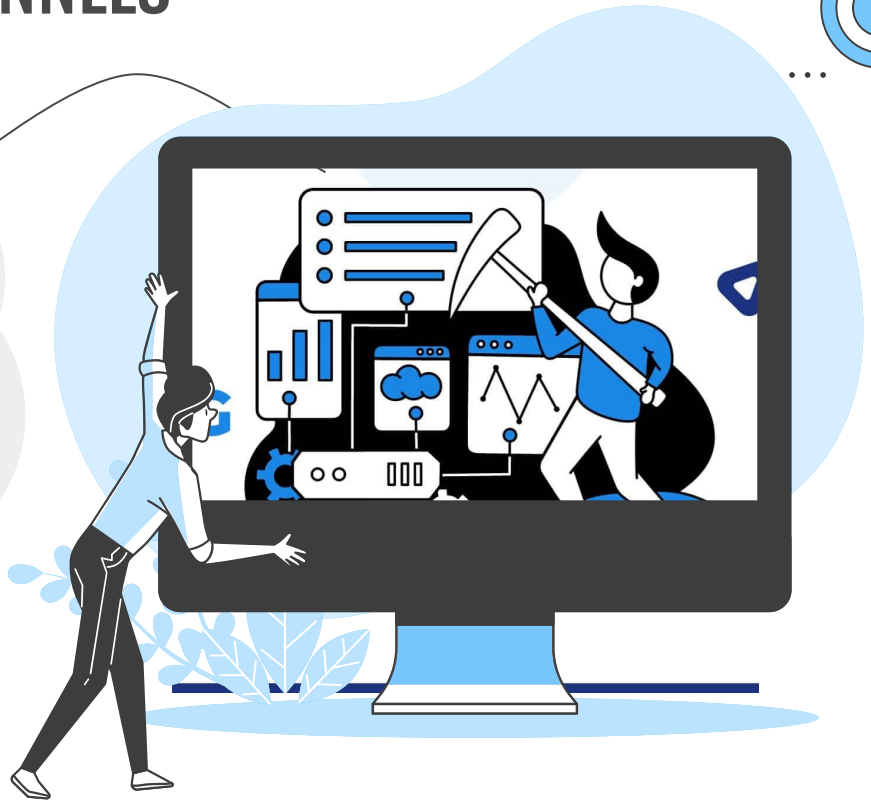
Dropping data

Supprimer les valeurs aberrantes

Remplacer les valeurs

Transformer les types des donnees

Supprimer les cellules vides



Suppression des valeurs textuelles

```
[ ] # Get the list of categorical columns
    categorical_cols = df.select_dtypes(include=['object']).columns.tolist()

    # Print the list of categorical columns
    print("Categorical Columns:")
    for col in categorical_cols:
        print(col)
```

```
Categorical Columns:
FILENAME
URL
Domain
TLD
Title
```

```
[ ] df= df.drop(columns=['FILENAME', 'URL', 'Domain', 'Title'])
```

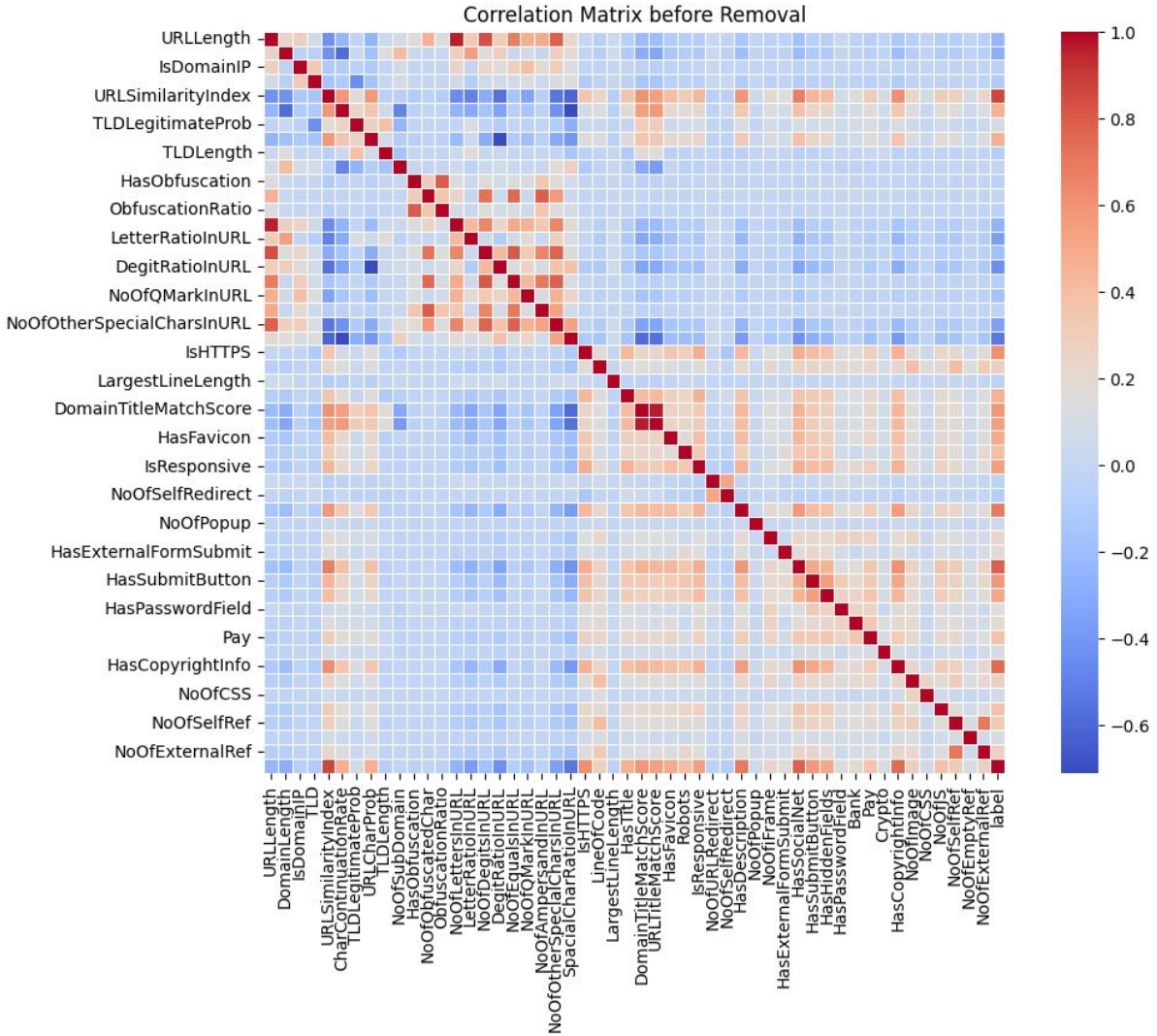
Transformation des valeurs textuelles a des valeurs numériques

```python
# Enumerate the values in the "TLD" column
df['TLD'], tld_enum = pd.factorize(df['TLD'])

# Print the enumerated values
print("Enumerated values for column 'TLD':")
print(df['TLD'])

# Print the unique values corresponding to the enumerated values
print("\nUnique values corresponding to the enumerated values:")
print(tld_enum)
```

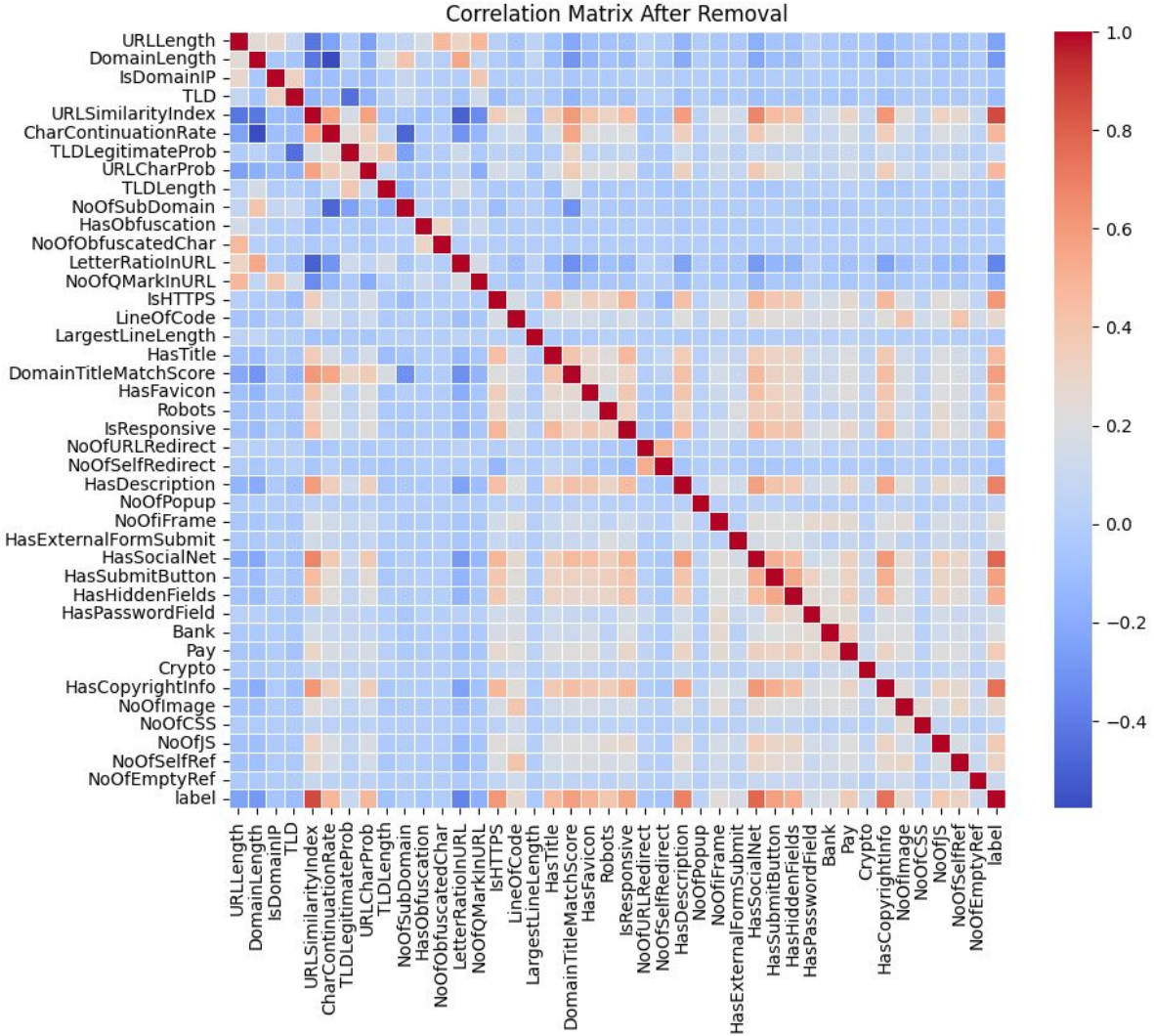
MATRICE DE CORRÉLATION ET SUPPRESSIONS DES VALEURS HAUTEMENTS CORRÉLÉS

```python
# Calculate the correlation matrix
correlation_matrix = df.corr('pearson')

# Set a threshold for correlation
threshold = 0.7    # value optimal between 0.7 & 0.9

print("Removed features:")

# Find pairs of features with correlation greater than the threshold
correlated_features = set()
for _ in range(4):
    correlation_matrix = df.corr('pearson').abs()
    for i in range(len(correlation_matrix.columns)):
        for j in range(i):
            if i != j and abs(correlation_matrix.iloc[i, j]) > threshold:
                colname = correlation_matrix.columns[i]
                if colname != "label":
                    correlated_features.add(colname)
                    print(colname)
    # Remove highly correlated features
    # Check if the columns exist in the dataframe before dropping them
    existing_columns = [col for col in correlated_features if col in df.columns]
    df = df.drop(columns=existing_columns)
```

AVANT SUPPRESSION DES VALEURS HAUTEMENTS CORRÉLÉES



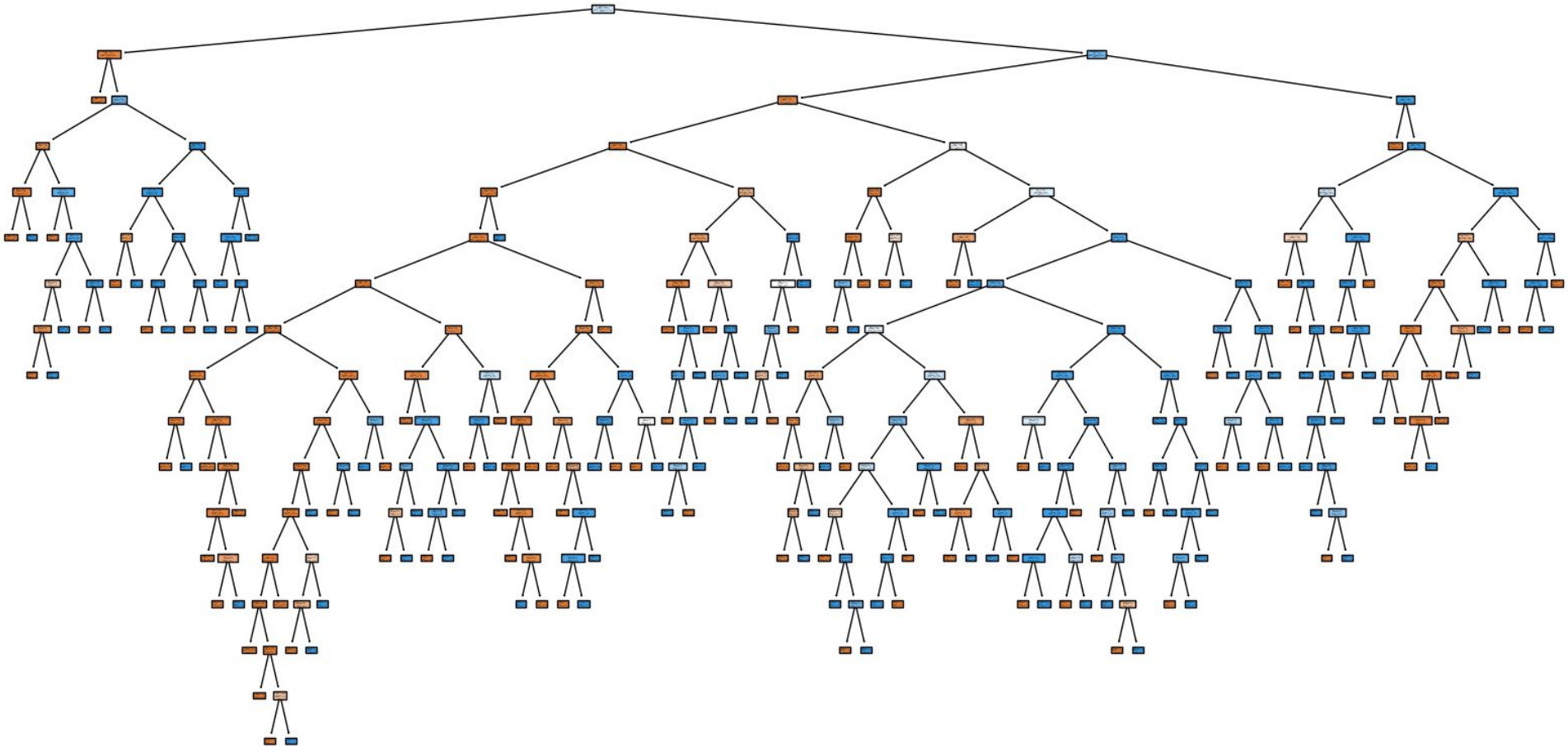
APRES SUPPRESSION DES VALEURS HAUTEMENTS CORRÉLÉES

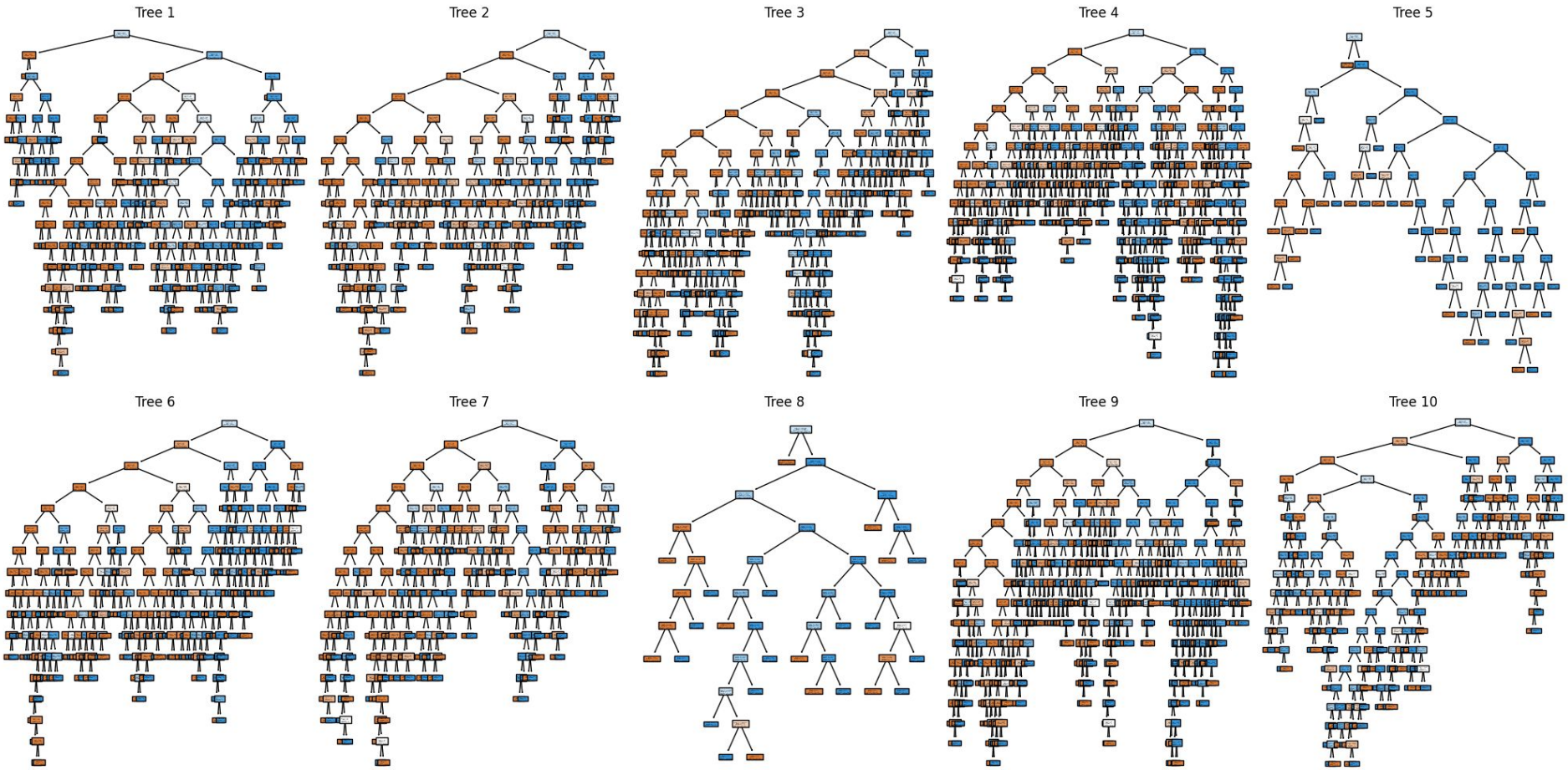


```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Creating and training the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(X_train, y_train)

# Making predictions on the testing set
y_pred = rf_classifier.predict(X_test)

# Calculating the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
```

First Decision Tree



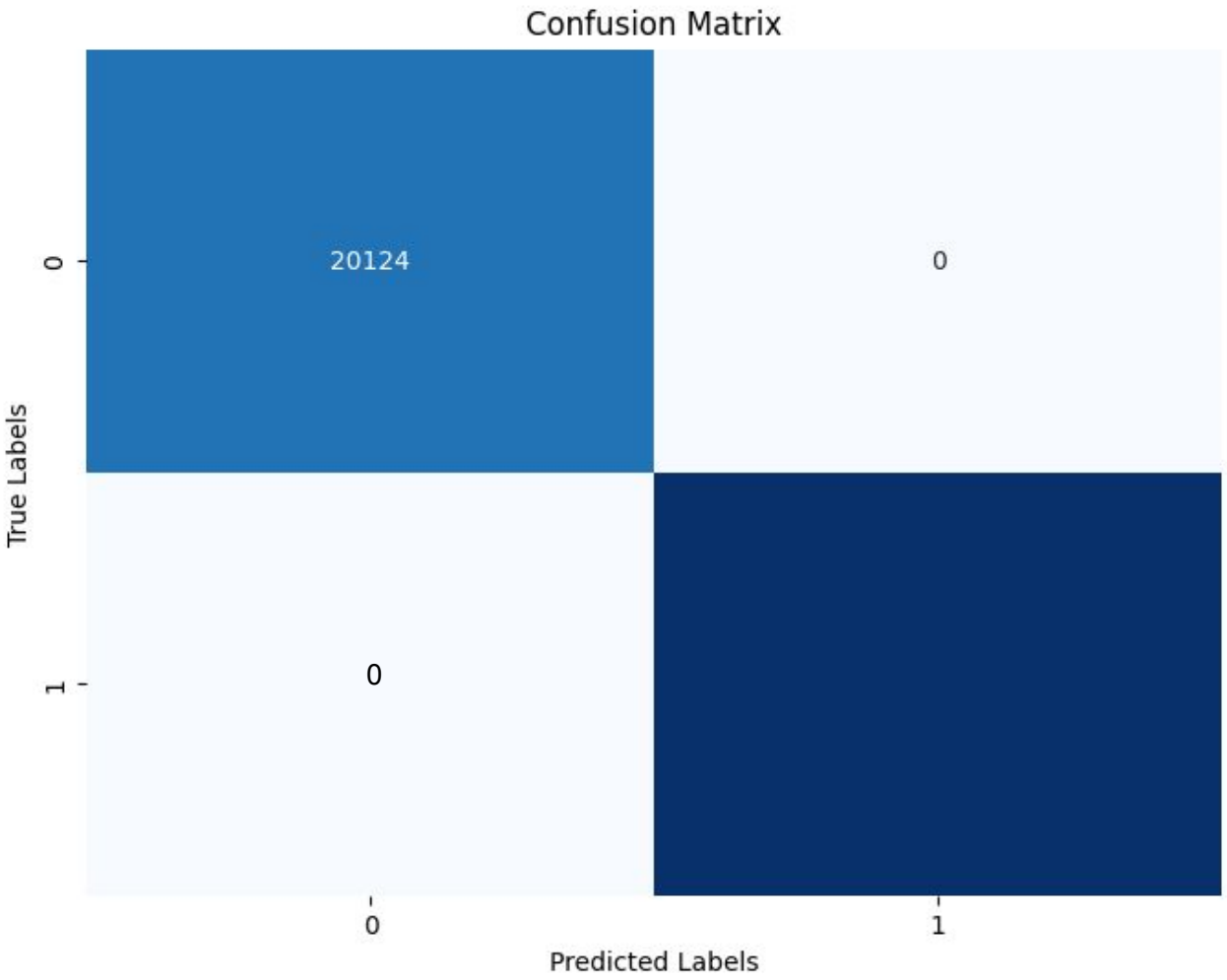

Tree 1
Tree 2
Tree 3
Tree 4
Tree 5
Tree 6
Tree 7
Tree 8
Tree 9
Tree 10

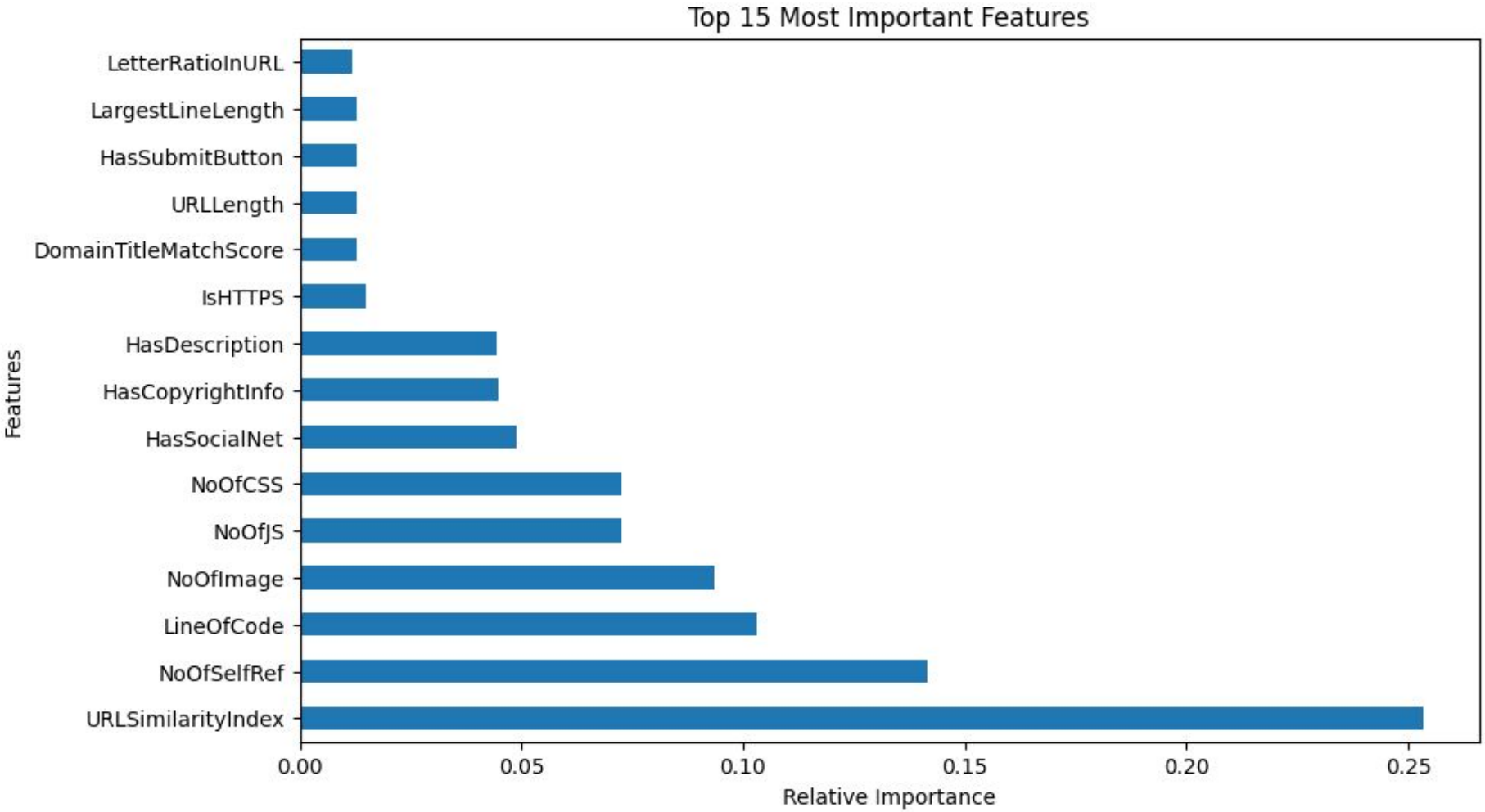
4

RESULTATS

Classification Report

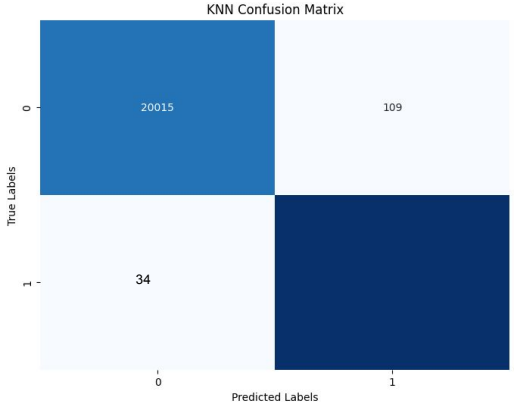
	Precision	Recall	F1-Score	Support
0	1.0000000	1.0000000	1.0000000	20214
1	1.0000000	1.0000000	1.0000000	27035
Macro-avg	1.0000000	1.0000000	1.0000000	47159
Weighted avg	1.0000000	1.0000000	1.0000000	47159
Accuracy	1.0000000		47159	

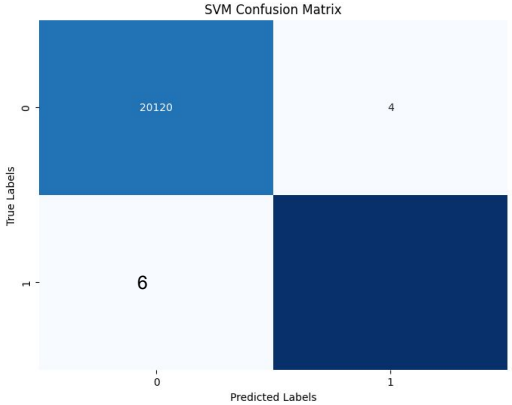

Confusion Matrix
20124
0
0
0
True Labels
0
1
0
1
Predicted Labels


Top 15 Most Important Features
LetterRatioInURL
LargestLineLength
HasSubmitButton
URLLength
DomainTitleMatchScore
IsHTTPS
HasDescription
HasCopyrightInfo
HasSocialNet
NoOfCSS
NoOfJS
NoOfImage
LineOfCode
NoOfSelfRef
URLSimilarityIndex
Features
0.00
0.05
0.10
0.15
0.20
0.25
Relative Importance

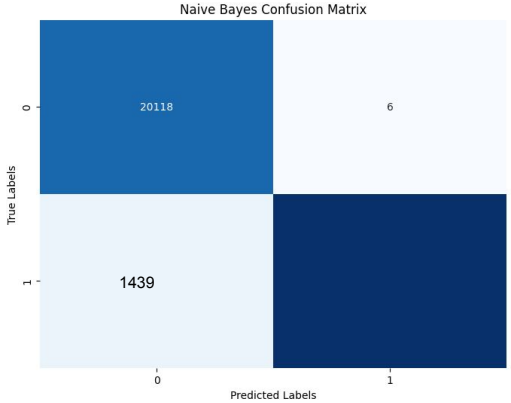
5

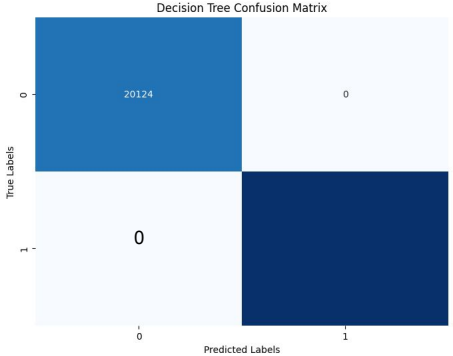
COMPARISON

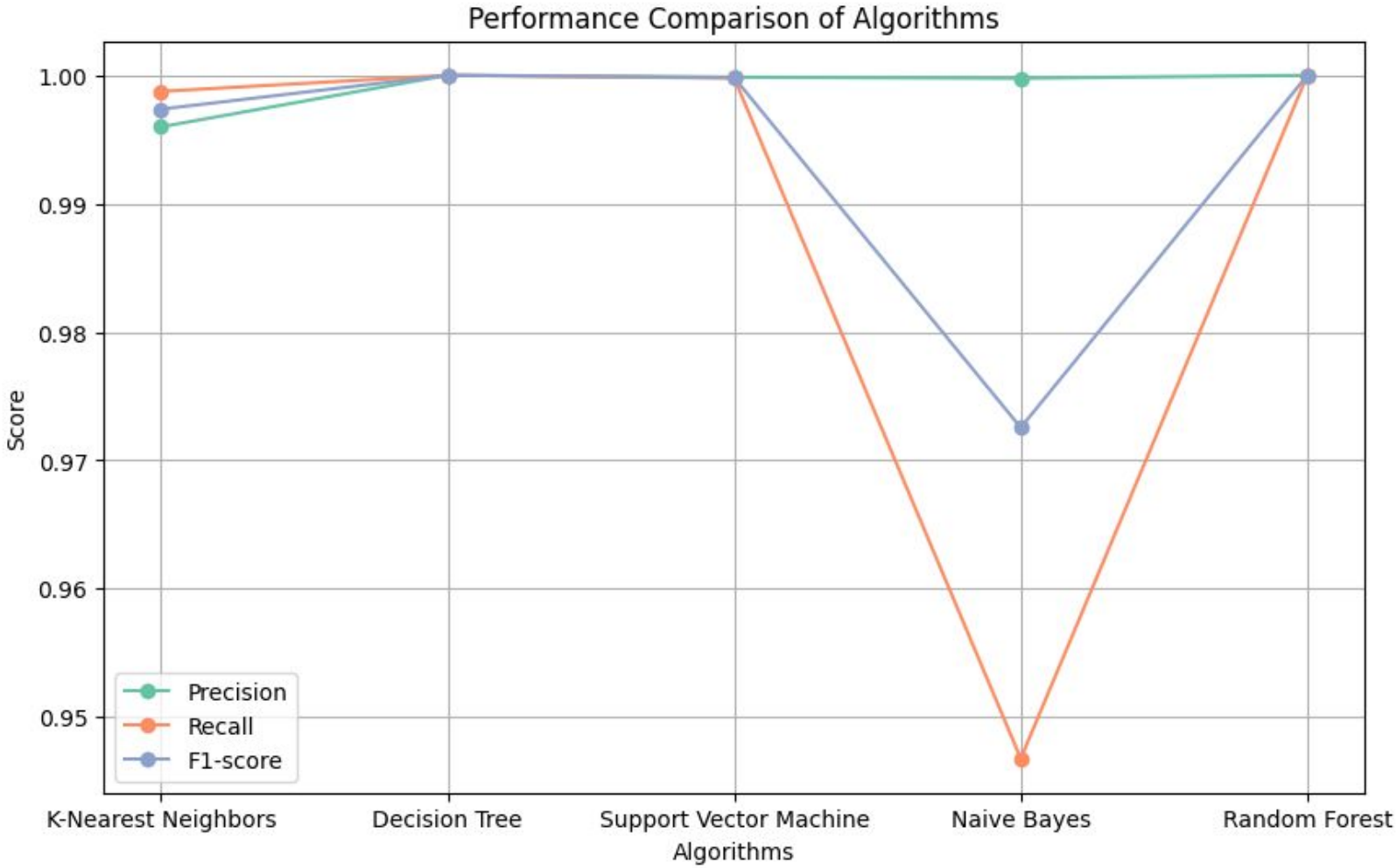

KNN Confusion Matrix
20015
109
34
True Labels
Predicted Labels
0
1


SVM Confusion Matrix
20120
4
6
True Labels
Predicted Labels
0
1

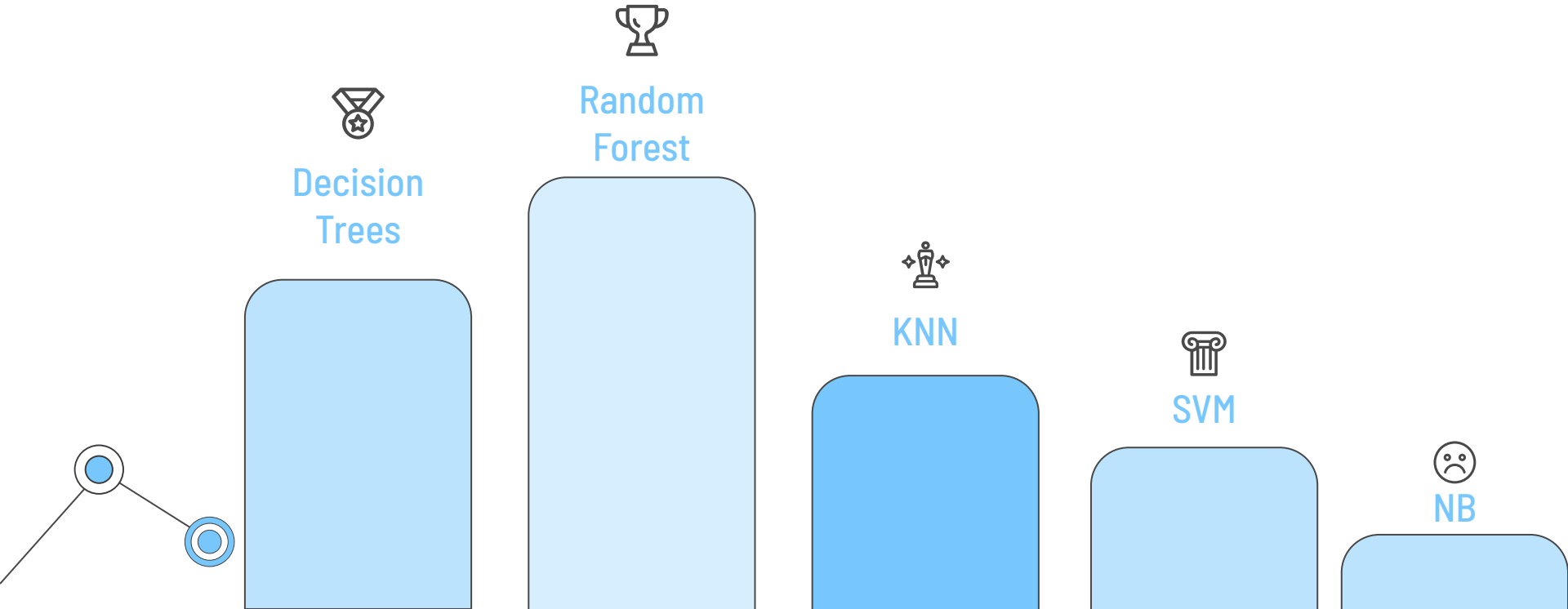

READING DATA
SPLITTING
TRAINING
TESTING


Naive Bayes Confusion Matrix
20118
6
1439
True Labels
Predicted Labels
0
1


Decision Tree Confusion Matrix
20124
0
0
True Labels
Predicted Labels
0
1


Performance Comparison of Algorithms
1.00
0.99
0.98
0.97
0.96
0.95
Score
Precision
Recall
F1-score
K-Nearest Neighbors
Decision Tree
Support Vector Machine
Naive Bayes
Random Forest
Algorithms

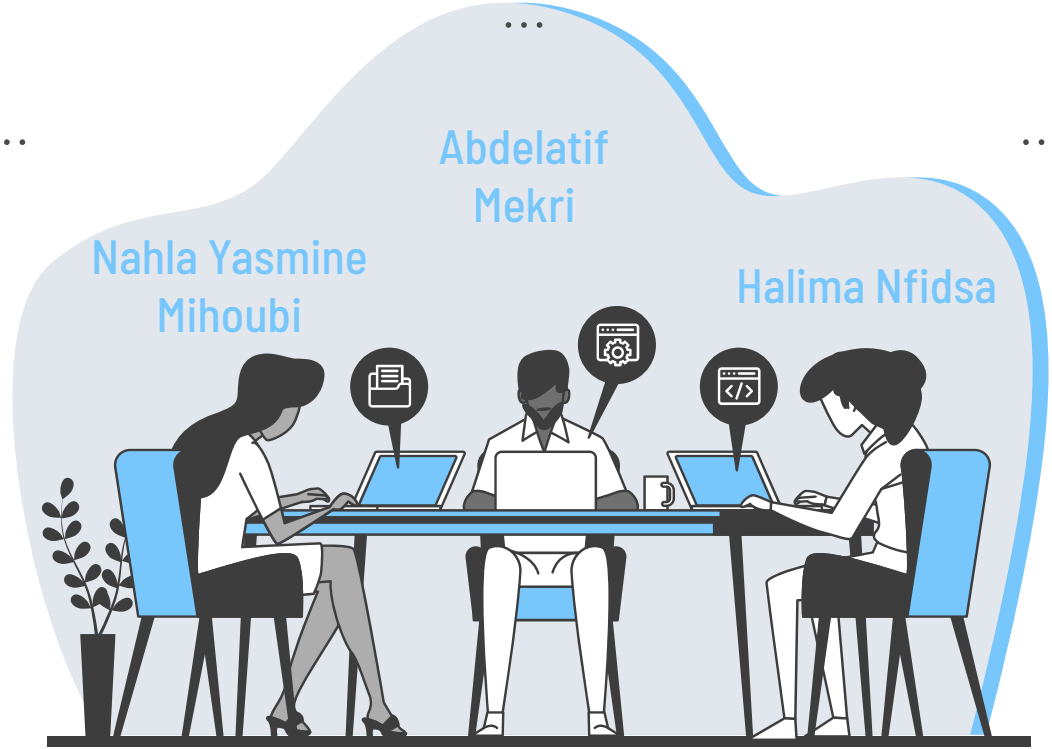
Awards



CONCLUSION

ce projet a démontré l'efficacité des techniques d'apprentissage automatique pour la détection des URLs de phishing, avec la forêt aléatoire se démarquant comme le modèle le plus performant.

TEAM



Merci Pour Votre Attention

DISCUSSION