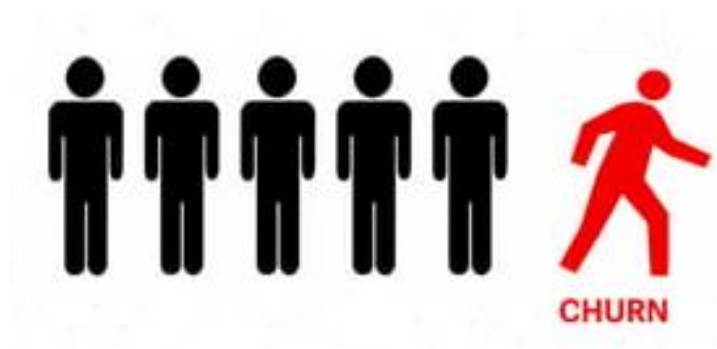


## Rapport de projet

# Prédiction de désabonnement des clients d'opérateur de Télécommunication en temps réel



Cycle Master en Sciences et Techniques

Filière : Sécurité IT et Big Data

Réalisé par :

- Abderrazzak EL BOURKADI
- Adnan EL HAYANI
- Ayoub ET-TOUBI
- Firdaous BOULBEN

Encadré par :

- Pr. Yasyn EL YUSUFI

*Année Universitaire : 2023/2024*

## 1. Introduction

Comprendre et anticiper le désabonnement des clients est devenu un enjeu majeur pour les opérateurs Telecom. La perte de clients représente non seulement une perte de revenus directe, mais aussi des coûts associés à l'acquisition de nouveaux clients. En effet, il est généralement plus coûteux d'acquérir un nouveau client que de fidéliser un client existant.

Face à ce défi, la prédiction précise du désabonnement des clients offre aux opérateurs Telecom un outil précieux pour identifier les clients à risque de quitter l'opérateur avant qu'ils ne le fassent réellement, et alors mettre en œuvre des stratégies de fidélisation ciblées pour retenir ces clients et réduire le taux de churn.

Dans ce contexte, ce projet présente une solution innovante basée sur le traitement de flux de données en temps réel pour prédire le désabonnement des clients Telecom. Ce rapport couvrira en détail l'architecture du système avec les choix technologiques, ainsi que les fonctionnalités et l'utilisation de l'application.

## 2. Architecture du Système

L'architecture du système repose sur une approche de traitement de flux de données en temps réel. Chaque composant de l'architecture est soigneusement sélectionné pour permettre une prédiction précise du désabonnement des clients Telecom. :

### 2.1. Apache Kafka Stream

Pour l'ingestion des données en temps réel, Apache Kafka est utilisé. Kafka offre une solution robuste pour collecter et gérer les flux de données provenant d'un jeu de données "churn-bigml".



### 2.2. Apache Spark

Utilisé pour le traitement des flux de données en temps réel. Spark permet de transformer, agréger et enrichir les données pour préparer les caractéristiques nécessaires aux modèles prédictifs.



### 2.3. [MongoDB](#)

Utilisé comme base de données NoSQL pour stocker les données clients traitées, les modèles et les résultats des prédictions.



### 2.4. [Flask](#)

Une application web développée avec Flask affiche les prédictions en temps réel. Les utilisateurs peuvent visualiser les tendances et les prédictions de désabonnement, et prendre des mesures proactives.



### 2.5. [Docker](#)

Utilisé pour containeriser les différents composants de l'application, assurant ainsi une portabilité et une cohérence entre les environnements de développement, de test et de production.



## 3. [Etapes d'implémentation](#)

### 3.1. [Ingestion des données](#)

Nous avons utilisé Apache Kafka Streams pour lire en temps réel les données du fichier "churn-bigml-80.csv". Kafka Streams nous permet de consommer les données du fichier CSV et de les transférer dans un flux Kafka pour un traitement ultérieur.

### 3.2. [Prétraitements des données](#)

Nous avons utilisé PySpark MLlib pour effectuer les prétraitements nécessaires sur les données du fichier. PySpark MLlib nous permet de manipuler et de transformer les données, notamment en les nettoyant, en les normalisant et en les préparant pour l'entraînement des modèles.

### 3.3. [Entraînement des modèles machine learning supervisés](#)

Nous avons entraîné trois modèles de machine learning supervisés sur la base d'apprentissage "churn-bigml-80.csv" :

- Support vector machine (SVM)
- Random Forest
- Gradient Boosting

Une fois les modèles entraînés, nous avons sélectionné "Gradient Boosting" comme le meilleur modèle et nous l'avons sauvegardé sous format .pkl pour une utilisation ultérieure dans la prédiction en temps réel.

### 3.4. Prédiction en temps réel

Nous avons utilisé le modèle pré-entraîné et sauvegardé pour prédire en temps réel si un client quittera ou non l'opérateur sur les données de test "churn-bigml-20.csv". Les prédictions sont effectuées à mesure que de nouvelles données arrivent, en utilisant Apache Kafka Streams pour la consommation des données et l'application du modèle.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

[appuser@kafka2 opt]$ python Producer.py
INFO:kafka.conn:<BrokerConnection node_id=bootstrap-0 host=localhost:9092 <connecting> [IPv4 ('127.0.0.1', 9092)]>: connecting to localhost:9092 [('127.0.0.1', 9092) IPv4]
INFO:kafka.conn:Probing node bootstrap-0 broker version
INFO:kafka.conn:<BrokerConnection node_id=bootstrap-0 host=localhost:9092 <connecting> [IPv4 ('127.0.0.1', 9092)]>: Connection complete.
INFO:kafka.conn:Broker version identified as 2.5.0
INFO:kafka.conn:Set configuration api_version=(2, 5, 0) to skip auto check_version requests on startup
INFO:root:Data sent to Kafka: {'State': 'LA', 'Account length': 117, 'Area code': 408, 'International plan': 'No', 'Voice mail plan': 'No', 'Number vmail messages': 0, 'Total day minutes': 184.5, 'Total day calls': 97, 'Total day charge': 31.37, 'Total eve minutes': 351.6, 'Total eve calls': 80, 'Total eve charge': 29.89, 'Total night minutes': 215.8, 'Total night calls': 90, 'Total night charge': 9.71, 'Total intl minutes': 8.7, 'Total intl calls': 4, 'Total intl charge': 2.35, 'Customer service calls': 1, 'Churn': False}
INFO:kafka.conn:<BrokerConnection node_id=2 host=127.0.0.1:9092 <connecting> [IPv4 ('127.0.0.1', 9092)]>: connecting to 127.0.0.1:9092 [('127.0.0.1', 9092) IPv4]
INFO:kafka.conn:<BrokerConnection node_id=2 host=127.0.0.1:9092 <connecting> [IPv4 ('127.0.0.1', 9092)]>: Connection complete.
INFO:kafka.conn:<BrokerConnection node_id=bootstrap-0 host=localhost:9092 <connected> [IPv4 ('127.0.0.1', 9092)]>: Closing connection.
INFO:root:Data sent to Kafka: {'State': 'IN', 'Account length': 65, 'Area code': 415, 'International plan': 'No', 'Voice mail plan': 'No', 'Number vmail messages': 0, 'Total day minutes': 129.1, 'Total day calls': 137, 'Total day charge': 21.95, 'Total eve minutes': 228.5, 'Total eve calls': 83, 'Total eve charge': 19.42, 'Total night minutes': 208.8, 'Total night calls': 111, 'Total night charge': 9.4, 'Total intl minutes': 12.7, 'Total intl calls': 6, 'Total intl charge': 3.43, 'Customer service calls': 4, 'Churn': True}
INFO:root:Data sent to Kafka: {'State': 'NY', 'Account length': 161, 'Area code': 415, 'International plan': 'No', 'Voice mail plan': 'No', 'Number vmail messages': 0, 'Total day minutes': 332.9, 'Total day calls': 67, 'Total day charge': 56.59, 'Total eve minutes': 317.8, 'Total eve calls': 97, 'Total eve charge': 27.01, 'Total night minutes': 160.6, 'Total night calls': 128, 'Total night charge': 7.23, 'Total intl minutes': 5.4, 'Total intl calls': 9, 'Total intl charge': 1.46, 'Customer service calls': 4, 'Churn': True}
INFO:root:Data sent to Kafka: {'State': 'SC', 'Account length': 111, 'Area code': 415, 'International plan': 'No', 'Voice
```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

24/05/23 14:37:18 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
24/05/23 14:37:18 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.ForeignLink
erBLAS
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

|State|Account length|Area code|International plan|Voice mail plan|Number vmail messages|Total day minutes|Total day call
s|Total day charge|Total eve minutes|Total eve calls|Total eve charge|Total night minutes|Total night calls|Total night c
harge|Total intl minutes|Total intl calls|Total intl charge|Customer service calls|Churn_prediction|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

|MN |81 |415 |0 |0 |0 |198.4 |93
|33.73 |210.9 |108 |17.93 |193.3 |71 |8.7
|10.4 |6 |2.81 |2 |True |
|SD |145 |408 |0 |0 |24 |147.5 |90
|25.08 |175.7 |108 |14.93 |252.1 |102 |11.34
|15.6 |3 |4.21 |2 |True |
|OK |89 |510 |0 |0 |0 |303.9 |95
|51.66 |260.9 |114 |22.18 |312.1 |89 |14.04
|5.3 |3 |1.43 |1 |True |
|CT |199 |415 |0 |0 |34 |230.6 |121
|39.2 |219.4 |99 |18.65 |299.3 |94 |13.47
|8.0 |2 |2.16 |0 |True |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

### 3.5. Sauvegarde des résultats

Les résultats des prédictions sont enregistrés dans une base de données MongoDB Les données de prédiction ont été stockées dans MongoDB pour faciliter la récupération et la présentation efficaces des données.

ChurnPrediction.Predictions 384 1  
DOCUMENTS INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

[EXPLAIN](#) [Reset](#) [Find](#) [</>](#) [Options](#)

[ADD DATA](#) [EXPORT DATA](#) [UPDATE](#) [DELETE](#) 1 - 20 of 384 [<](#) [>](#) [≡](#) [{ }](#) [田](#)

```

{
  "_id": {
    "$oid": "664f5fd89774bb21b388e143"
  },
  "State": "MI",
  "Account length": 51,
  "Area code": 415,
  "International plan": 0,
  "Voice mail plan": 0,
  "Number vmail messages": 0,
  "Total day minutes": 229.6999969482422,
  "Total day calls": 129,
  "Total day charge": 39.04999923706055,
  "Total eve minutes": 336,
  "Total eve calls": 104,
  "Total eve charge": 28.559999465942383,
  "Total night minutes": 192.8000030517578,
  "Total night calls": 128,
  "Total night charge": 8.680000305175781,
  "Total intl minutes": 9.600000381469727,
  "Total intl calls": 1,
  "Total intl charge": 2.5899999141693115,
  "Customer service calls": 1,
  "Churn_prediction": "True"
}

```

```

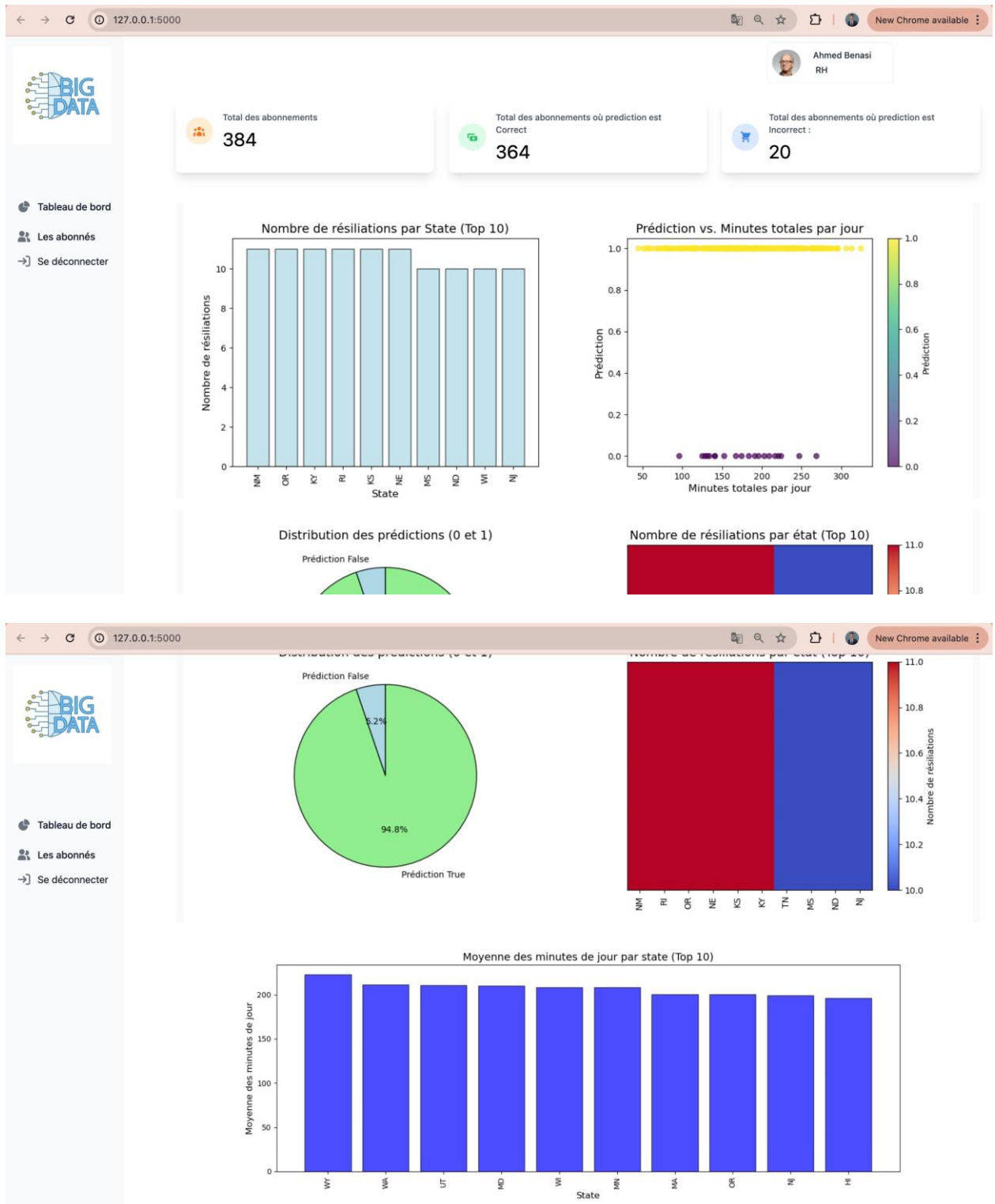
{
  "_id": { },
  "State": "ID",
  "Account length": 166,
  "Area code": 415,
  "International plan": 0,
  "Voice mail plan": 0,
  ...
}

```

### 3.6. Développement de l'Application Web

Nous avons développé une application web pour exploiter les données de test et présenter les résultats de traitement du churn en utilisant Flask.

## 4. Présentation de l'interface web



## **5. Conclusion**

Ce projet révolutionne la gestion de la clientèle dans le secteur des télécommunications en offrant une approche proactive et prédictive pour réduire le désabonnement des clients. L'application web, alimentée par l'analyse de données en temps réel et l'apprentissage automatique, permet aux opérateurs Telecom de prendre des décisions éclairées, d'améliorer l'expérience client et de stimuler la fidélisation à long terme.