REPORT OF IMPLEMENTATION

State of the art

Le réseau de détection de la qualité de l'air d'AirQo compte plus de 120 appareils à faible coût déployés à travers l'Ouganda ; dans la plupart des cas, ces appareils sont déployés dans des environnements non surveillés ou dangereux. Ces appareils électroniques à faible coût sont susceptibles de tomber en panne en raison d'un dysfonctionnement de la communication, du vieillissement, de l'usure, des défauts de fabrication, d'un étalonnage incorrect, d'une mauvaise manipulation et d'autres facteurs environnementaux externes. Les pannes entraînent des inexactitudes et des pertes de données, ce qui a un impact sur les décisions et les politiques qui pourraient avoir un impact significatif sur la vie des gens.

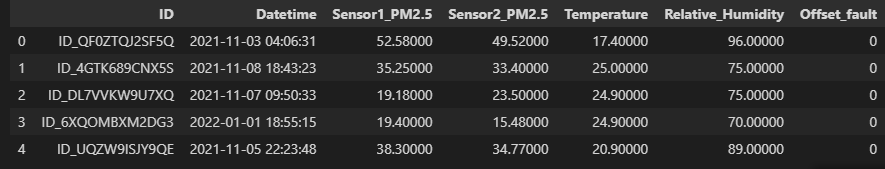
La détection et la surveillance des pannes d'appareils sont essentielles pour les activités d'AirQo ; les appareils défectueux doivent être identifiés, isolés et réparés ou remplacés de toute urgence. Les données reçues d'un appareil peuvent être utilisées pour déterminer si l'appareil fonctionne correctement ou non.

C’est ainsi que notre tâche consiste à développer un modèle de classification pour identifier un appareil présentant ou non un défaut de décalage, quel que soit l'appareil. Le modèle peut être utilisé par AirQo pour signaler automatiquement un appareil qui renvoie des données erronées.

AirQo est un projet de recherche du Makerere University College of Computing and Information Sciences. Nous nous engageons à utiliser la technologie pour résoudre les problèmes sociaux à travers l'Afrique en mettant l'accent sur la qualité de l'air. Avec le soutien de Google, nous avons développé un réseau d'appareils à faible coût à travers l'Ouganda et utilisons ML/AI pour créer des données de prévision, spatiales et maintenant d'étalonnage. Nous partageons nos idées sur notre plateforme, notre application et notre site Web dans le but de sensibiliser aux causes et aux conséquences de la mauvaise qualité de l'air et de fournir des informations exploitables basées sur des données pour réduire la pollution de l'air en Ouganda et dans d'autres villes africaines.

About Dataset

Le dataset utilisé a été prit sur la plateforme [Zindi](https://zindi.africa/competitions/umojahack-africa-2022-beginner-challenge/data) qui a servi pour une compéttion sur la prédiction des défauts sur les capteurs de qualité d’air.Ce jeu de donné est constitué de deux fichiers, un pour l'entraînement( Train.csv) contenant 297177 lignes et 7 colones et un autre pour le test(Test.csv) contenant 127361 lignes et 6 colonnes. Le fichier train contient environ 300 000 lectures et le test contient environ 100000 lectures. Dans chaque fichier nous avons les colonnes suivantes :'Sensor1\_PM2.5', 'Sensor2\_PM2.5', 'Temperature', 'Relative Humidity', 'Datetime\_day', 'Datetime\_month', 'Datetime Year', 'Datetime Hour'



*Figure - Les 5 premières lignes du DataSet*

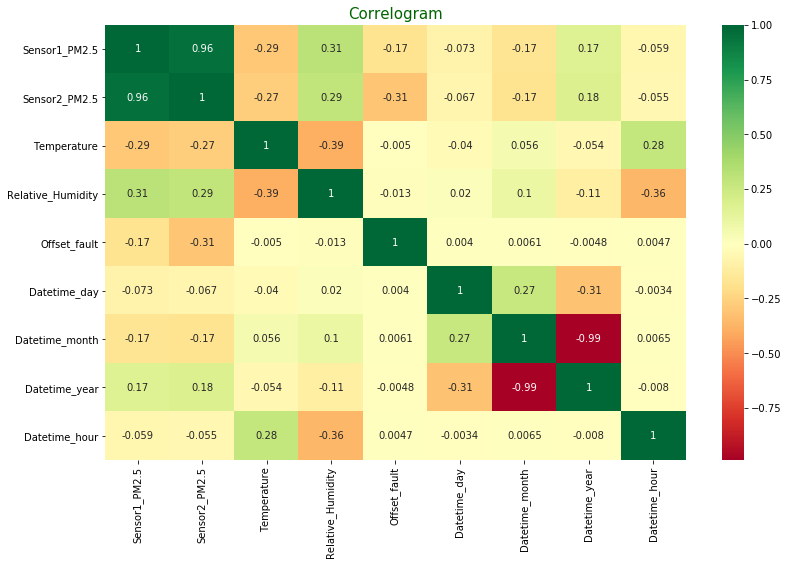
Notre dataset contient de nombreuses données manquantes et des valeurs abérantes alors avant d’entrainer le modèle il faudra faire une parfaite analyse exploratoire des données puis les infliger un traitement afin de les preparer pour la phase de modélisation.

Méthodologie

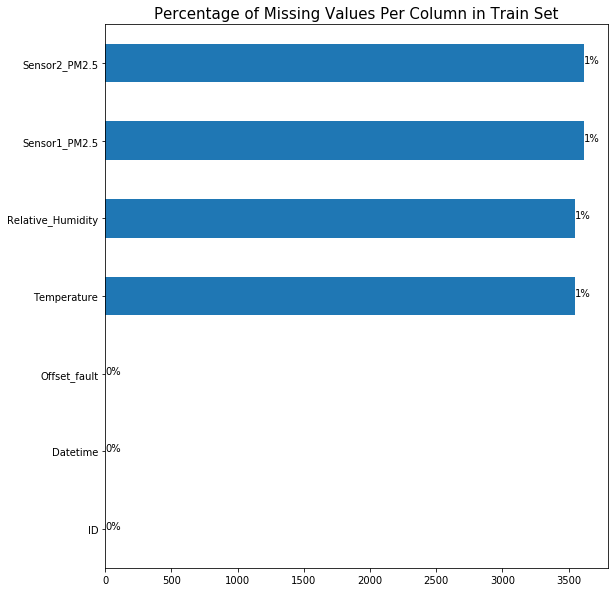
Pour être sûre que notre modèle soit bien construit et fonctionne correctement nous sommes passés par plusieurs étapes:

1- Faire une EDA(Exploratory Data Analysis) sérieuse en cherchant les variables significatives aussi

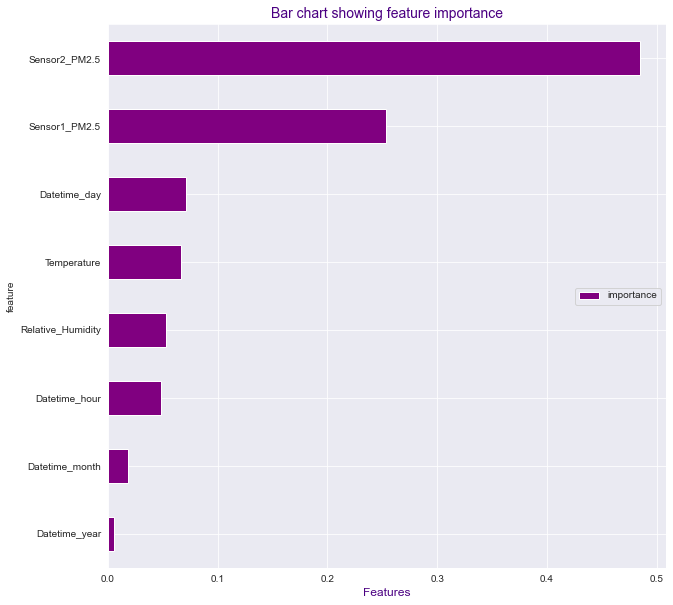
-identification de la corrélation des variables de corrélation



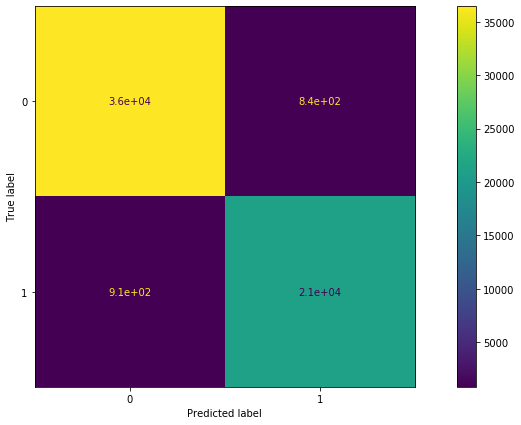
*Figure - Matrice de corrélation des variables*

**

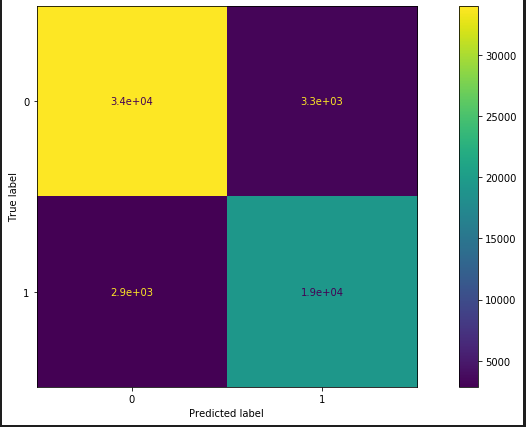
*Figure -pourcentage des valeurs manquantes par colonne*

**

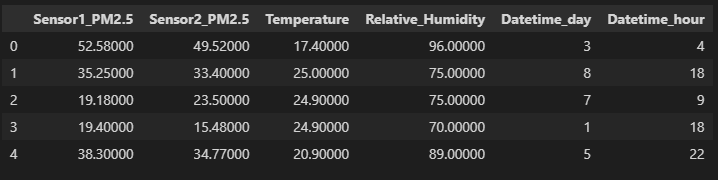
*Figure - variables importantes*

**

*Figure - Matrice de confusion de random forest*



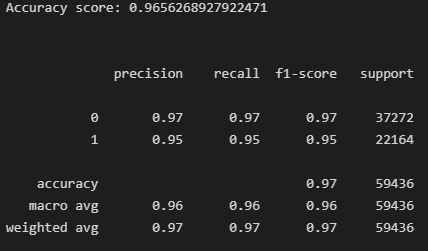
*Figure - Matrice deconfusion de svm*



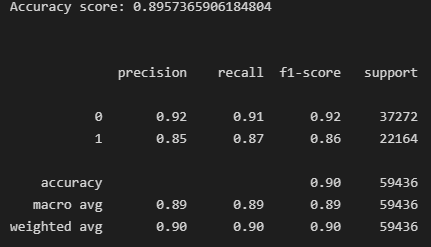
*Figure - dataset apres extraction des variables importantes*

2- Creation d’un jeu de donnée d'entraînement et de validation

3- Formation et évaluation du modèle



*Figure - score obtenu avec random forest*

**

*Figure -Score obtenu avec SVM*

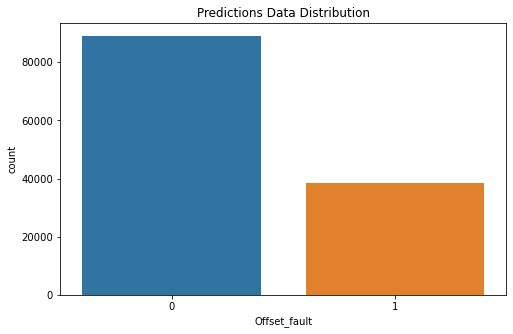
Résultat

1. Dans la phase d’exploration nous avons trouvé:

* 3614 valeurs NaN dans la variable Sensor1\_PM2.5
* 3614 valeurs NaN dans la variable Sensor2\_PM2.5
* 3549 valuers NaN dans la variable Temperature
* 3549 dans la variable Relative\_humidity
* Une forte corrélation positive de 0.966 est observée entre les variables Sensors1 PM2.5 et Sensorr2 PM2.5 et une autre fortement faible de -0.99 entre la Température et l’humidité
* la colonne Datetime a été décomposée en morceau de table de jour, mois et année et heure)

1. Dans la phase de modélisation

* Nous avons utilisé l’algorithme Random Forest qui a été évalué suivant la métrique ‘Accuracy’ et on a obtenu un score de 0.97 en un temps de calcul très court
* Nous avons également utilisé l’algorithme SVM pour classification soit SVC qui a été évalué suivant la métrique ‘Accuracy’ et on a obtenu un score de 0.89 en un temps de calcul très court
* On observe également dans les prédictions une forte tendance dans la variable 0 ce qui explique le déséquilibre de notre dataset qu’on pourra gérer dans un autre cadre



Requirements

* numpy
* matplotlib
* pandas
* seaborn
* sklearn

Conclusion

Les tâches de classification peuvent être gérées de diverses façons grâce à des algorithmes de classification mais alors il faut savoir déjà faire une bonne EDA et aussi savoir quel algorithme marche le mieux dans le dit problème. dans notre cas on peut dire que Random Forest a une meilleure qualité de performance sur les SMS en tout globalement.