Sorbonne Université

ue reds

Rapport TP 1 : Higgs Boson Machine Learning Challenge

Salomé Attiach, Yannis Karmim

Table des matières

1 Description et formalisation de la tâche

Pour ce projet dans le cadre de l'UE REDS (Research in Data Science and Methodology), on s'intéresse au projet Kaggle sur la détection du Boson de Higgs.
Les données ont été construites à partir d'un simulateur de détection de la particule du Boson de Higgs. Dans nos données on dispose d'événements, qui ont résultés ou non, de la détection du Boson de Higgs.
Pour un événement donné on a les informations sur les 30 paramètres de l'expérience (nos "features") ainsi que le résultat de cette expérience/simulation, si oui ou non le Boson de Higgs a été détecté.
Le label s signifie que l'on a bien reçu un signal du Boson de Higgs. Le label b signifie que l'on a pas détecté la particule.
Formellement, on se place dans une tâche de Machine Learning de **classification binaire** par apprentissage supervisé, puisque l'on dispose de données de train fournies par le challenge.

Maintenant que notre tâche est bien définie, l'exploration des données va nous permettre de :

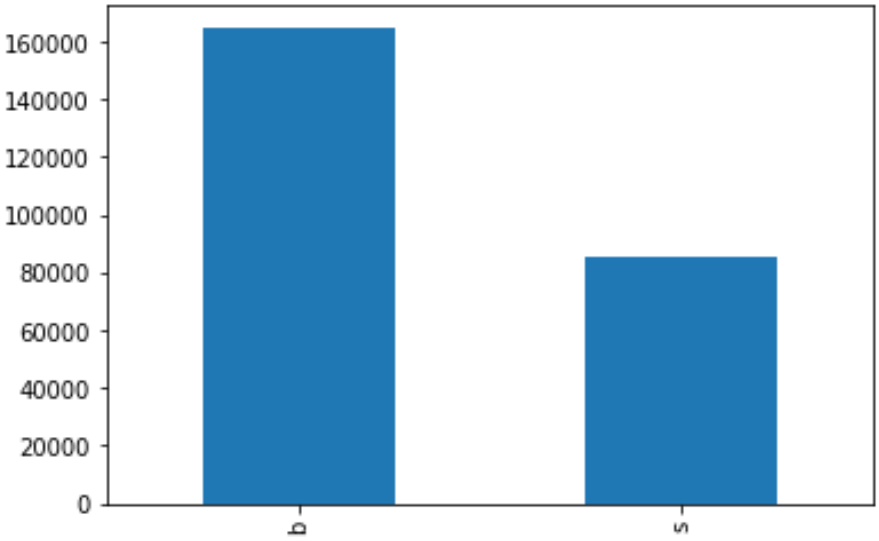
- Effectuer des analyses statistiques sur nos paramètres et nos labels (variance, moyenne...).
- Soulever les différentes difficultés liées aux données (classe déséquilibrée, données manquantes...)
- Analyser l'importance de certains paramètres, et chercher des possibles corrélation dans nos données.

Cette étape est essentielle pour bien cerner notre tâche pour que par la suite on puisse choisir nos modèles et définir une campagne d'expérience afin d'évaluer nos performances.

2 Exploration des données

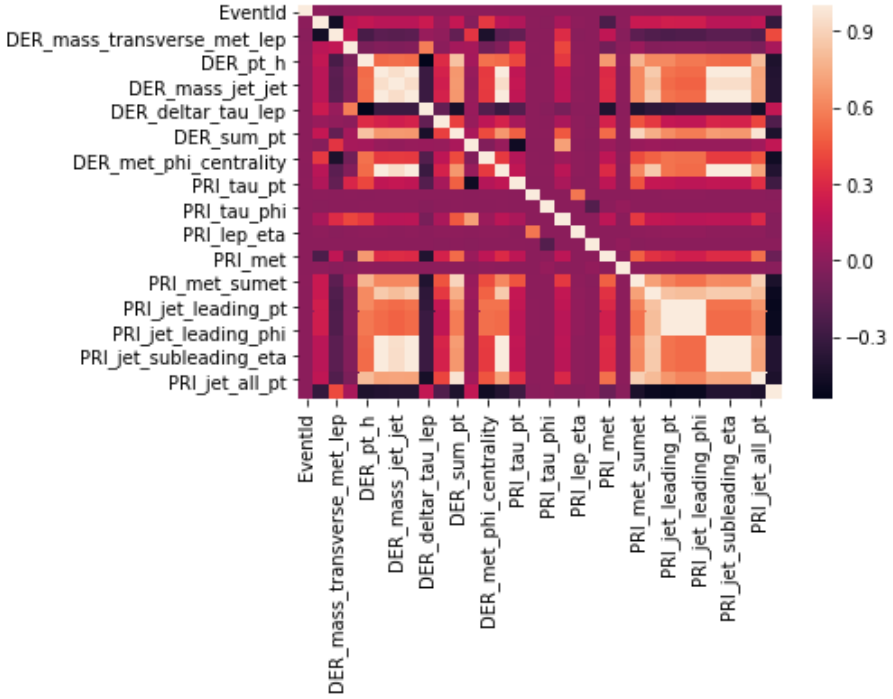
2.1 Répartition des classes.

Nous commençons notre exploration en regardant le pourcentage de chacune des classes, pour savoir si notre jeu de données est équilibré. Nous disposons de 250000 exemples d'évènements. Nous nommerons par la suite s et b les deux classes de notre problème de classification binaire.
Nous regardons donc le pourcentage de chaque classe dans notre jeu et nous obtenons les résultats suivants :
Pourcentage de la classe s : 34.3 %
Pourcentage de la classe b : 65.73 %
On représente ça sur un histogramme, qui confirme les pourcentages précédents :



2.2 Analyse des features

Chacun de nos évènements étant défini par 30 features, nous allons maintenant chercher à déterminer quelles sont les features les plus importantes, c'est à dire celles qui nous rapporterons le plus d'informations utiles à la classification.
Avant de chercher l'importance de ces features nous avons effectuer une matrice de corrélation afin de déterminer s'il existe des liens entre nos variables :



Cette matrice nous permet de savoir quelles paramètres sont plus ou moins corrélés, plus la couleur est claire plus la corrélation est forte.

Nous avons par la suite décider d'utiliser le classifieur Random Forest qui nous permet de déterminer la probabilité d'importance de chaque feature. Appliqué à notre jeu de données, on obtient la distribution de probabilité, qui nous permet par la suite de dire que les paramètres les plus importants sont les suivants :
DER_mass_MMC, DER_mass_transverse_met_lep, DER_mass_vis, DER_deltar_tau_lep, DER_pt_ratio_lep_tau, DER_met_phi_centrality, PRI_tau_pt, PRI_met

2.3 Analyse des données manquantes

Les données manquantes peuvent être un réel problème supplémentaire pour notre tâche. Elles sont représentées par la valeur **-999.0** dans notre dataset.
En analysant le nombre de lignes contenant des données manquantes on trouve qu'il y a environ **73%** des lignes qui en contiennent. Il est donc difficile de se passer de toutes les observations avec des informations manquantes pour notre apprentissage.

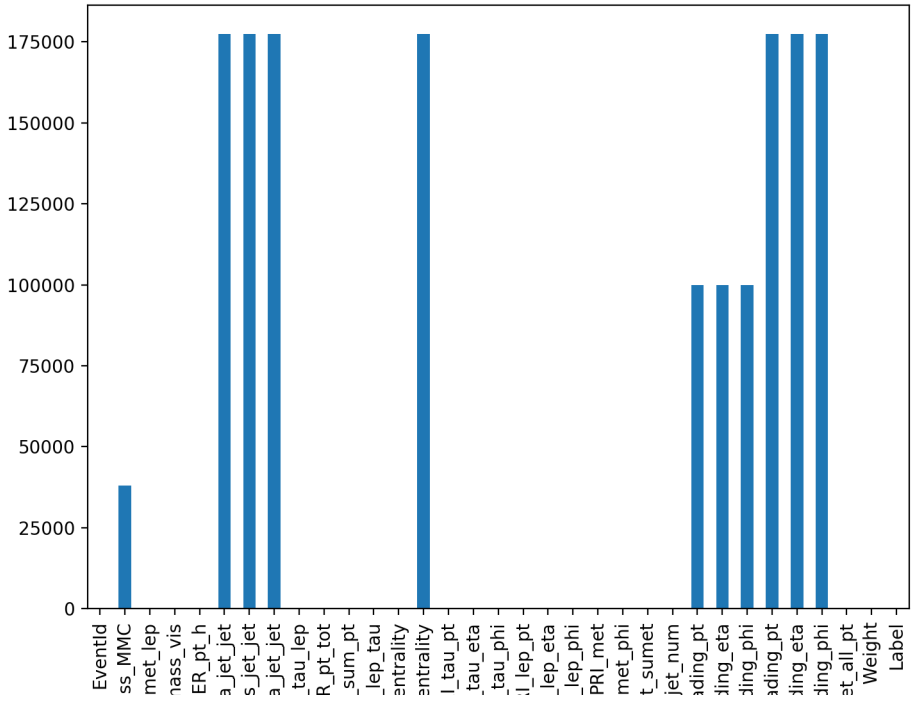


FIGURE 1 – Nombre de valeurs manquantes pour chaque features.

Par la suite il est crucial de se poser la question et d'expérimenter, si oui ou non, on garde nos features qui contiennent énormément de données manquantes, et comment traiter ces cas. On pourra par exemple expérimenter si l'on obtient des meilleures scores en supprimant les features concernées.