

Soutenance Finale

Projet Big Data

Sommaire

1. Présentation du sujet
2. Infrastructure
3. Analyse des données
4. Modèles de Machine Learning
5. Application Web et démonstration
6. Conclusion

Compréhension du sujet

Double Objectif : *Analyse de données & Proposition d'un modèle pour la prédiction du prix de la nuitée*

Etape 1

Les données sont rapatriées en local depuis HDFS.

Etape 2

Les données sont poussées sur une VM dans le cloud AWS de manière sécurisée (chiffrée).

Etape 3

Un modèle d'apprentissage est appris sur cette VM dans le cloud AWS.

Etape 4

Avec une partie des données, le modèle appris est exécuté sur la VM AWS pour créer un fichier predict.csv. Ce fichier sera sauvegardé sur le FileSystem de la VM AWS, au format CSV.

Etape 5

Les résultats predict.csv sont alors récupérés et chargés dans une base NoSQL (MongoDB) s'exécutant sur une autre VM.

Etape 6

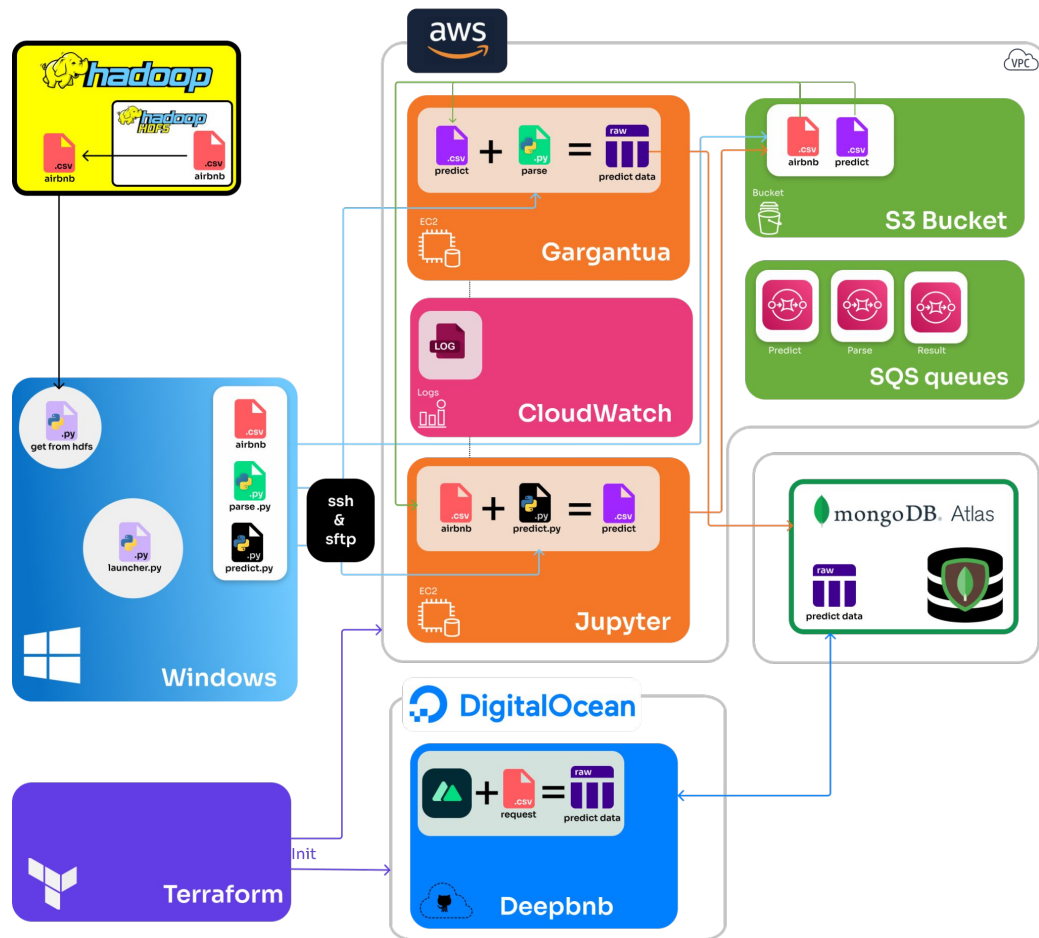
Une technologie de DataViz est utilisée afin d'afficher graphiquement les résultats (tout ou partie) et d'appuyer l'analyse et le message à faire passer.

100 % cloud

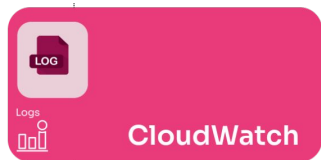
1 commande pour les contrôler tous

0 config*

Infrastructure



Infrastructure



aws Services Rechercher des services, des fonctions, des blogs, des documents et [Option+S]

CloudWatch

Favoris

Tableaux de bord

Alarmes

En alarme

Toutes les alarmes

Facturation

Journaux

Groupe de journaux

Logs Insights

Métriques

Toutes les métriques

Explorer

Flux

Traces X-Ray

Carte des services

Suivis

Événements

Règles

Bus d'événements

Surveillance des applications

Carte ServiceLens

État des ressources

Scripts Canary Synthetic

Évidement

Commentaires Français

CloudWatch > Log groups > big-data > ip-172-31-92-252.ec2.internal/parse_data.py/root/3574

Événements de journaux

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

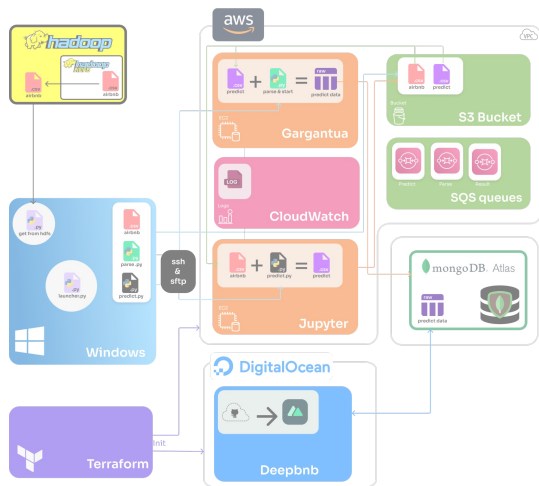
View as text Actions Create Metric Filter

Filtrer les événements

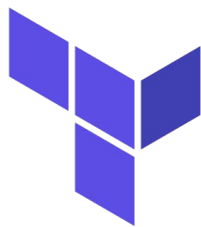
Clear 1m 30m 1h 12h Custom

Horodatage	Message
	Aucun ancien événement pour le moment Réessayer
2022-02-11T01:11:56.797+01:00	Application is starting
2022-02-11T01:38:52.806+01:00	New messages in ParseQueue
2022-02-11T01:38:52.871+01:00	File 2022-02-11T00:38:52_prediction.csv downloaded
2022-02-11T01:38:52.876+01:00	Inserting database...
2022-02-11T01:38:53.562+01:00	Inserting completed
2022-02-11T01:38:53.562+01:00	Deleting 2022-02-11T00:38:52_prediction.csv
2022-02-11T01:38:53.563+01:00	['6205b01ca626d11c4f5fb215']
2022-02-11T01:38:53.612+01:00	Message sent {'MD50FMessageBody': '8ea62084ca7e541d918e823422bd82e', 'MD50FMessageAttributes': '8afbdb9fbc896fca09615e96df258d', 'MessageId'...
2022-02-11T01:38:53.613+01:00	Task complete !
	Aucun nouvel événement pour le moment Nouvelle tentative Botanique suspendue

© 2022, Amazon Web Services, Inc. ou ses affiliés. Confidentialité Conditions Préférences relatives aux cookies



Automatisation des étapes



HashiCorp

Terraform

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }

  required_version = ">= 0.14.9"
}

provider "aws" {
  profile = "default"
  region = "us-east-1"
}

resource "aws_security_group" "ssh-sg" {
  name = "ssh-sg"
  ingress {
    from_port = 22
    to_port   = 22
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
  egress {
    from_port = 0
    to_port   = 443
    protocol = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }
}

resource "tls_private_key" "ssh_key" {
  algorithm = "RSA"
  rsa_bits = 4096
}
...
```

→ iac-in-action git:(main) ×

```
{
  "gargantua_server": {
    "sensitive": false,
    "type": "string",
    "value": "ec2-34-227-57-172.compute-1.amazonaws.com"
  },
  "jupyter_server": {
    "sensitive": false,
    "type": "string",
    "value": "ec2-54-225-48-21.compute-1.amazonaws.com"
  },
  "ssh_key": {
    "sensitive": true,
    "type": [
      "object",
      ...
    ]
  }
}
```

Analyse des données

Retrait de colonnes : - Identifiant

- Url
- Titre
- Résumé
- Nombre de lit
- Nombre de salle de bain
- Type de lit
- Résumé
- Animal sur place
- Règlement intérieur
- Prix nuité

Retrait de lignes : - Prix nuitée à 0

- Descriptions manquantes
- Type de propriété inconnue

Changement de nom des colonnes selon la norme snake_case

Analyse des données textuelles

Trois colonnes avec des catégories : **type de logement**, **type de propriété**, **conditions d'annulation**

Type de logement:

- Chambre partagé
- Chambre privée
- Logement entier

`partagée < privée < logement entier`

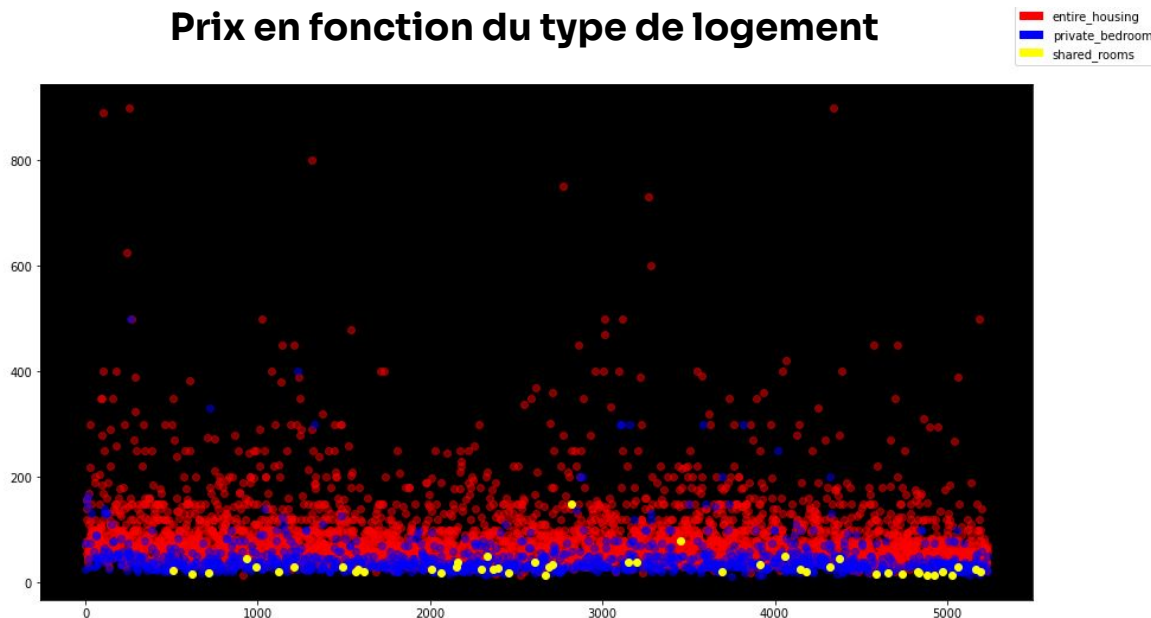
Pas de différence fixe entre les catégories

Catégories de différent cardinal

One-hot encoder

Encodage sur 4 colonnes

Prix en fonction du type de logement



Analyse des données textuelles

Trois colonnes avec des catégories : **type de logement**, **type de propriété**, **conditions d'annulation**

Type de propriété:

- Maison
- Appartement
- Villa
- ...

Pas d'ordre de grandeur entre les catégories

Beaucoup de catégories (12)

- house (Maison)
- apartment (Appartement)
- bed_and_breakfast (Bed & Breakfast)
- city_house (Maison de ville)
- loft (Loft)
- cabin (Cabane)
- apartment_in_residence (Appartement en résidence)
- bungalow (Bungalow)
- eco_house (Maison écologique)
- villa (Villa)
- dormitory (Dortoir)
- other (Autre)

Binary encoder

Encodage sur 4 colonnes

Analyse des données textuelles

Trois colonnes avec des catégories : **type de logement**, **type de propriété**, **conditions d'annulation**

Conditions d'annulation :

- Aucune condition - 1
- Flexible - 2
- Modérées - 3
- Strictes - 4

Ordre de grandeur entre les catégories

Interval similaire entre les catégories

- `none` (nan : Aucune condition)
- `flexible` (Flexibles)
- `moderate` (Modérées)
- `strict` (Strictes)

Ordinal encoder

Encodage sur 1 colonne

Analyse des données textuelles

Choix Préalables

- Pas de traitement des langues étrangères (tentatives non-concluantes)
- Traitement de la description de l'annonce

Extraction de Features

- TF-IDF avec sélection par le χ^2
- Extraction des features les plus caractéristiques pour chaque prix

Pré-Traitements

- Isolation des mots en minuscule
- Suppression des “stop words”

Insertion dans le Jeu de Données

- Test de la présence de la feature dans la description
- Alimentation du “dataframe” du modèle

Modèles de Machine Learning

Random Forest Regressor

Module scikit-learn

<https://larevueia.fr/regression-avec-random-forest-predire-le-lover-dun-logement-a-paris/>

Méthode d'assemblage d'arbres de décision indépendants

Principe : Tree Bagging + Feature Sampling

Avantages :

- Intuitif à comprendre
- Rapide à entraîner

Inconvénient :

- Boîte noire qui rend l'explication des résultats difficiles

XGB Regressor

Hors scikit-learn

<https://datascientest.com/xgboost-grand-gagnant-des-competitions-machine-learning-algorithme>

XGB = eXtreme Gradient Boosting

Principe : Boosting par appels successifs d'algorithme peu performants et utilisation des résidus successifs

Avantages :

- Facile à intégrer avec scikit-learn
- Performant & Paramétrable

Inconvénient :

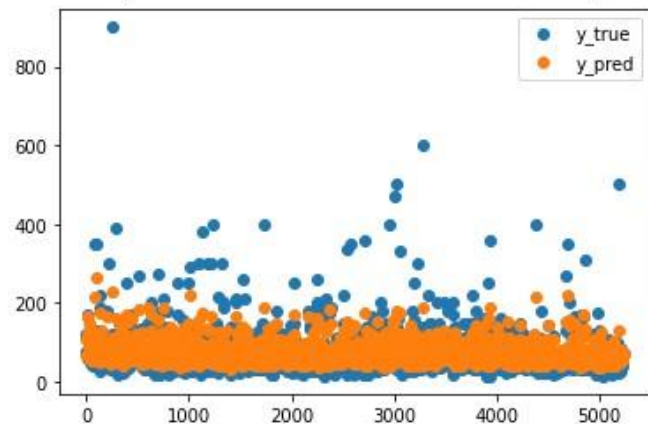
- Sensible à l'over-fitting

Modèles de Machine Learning

Random Forest Regressor

```
rf.score(X_test,y_test) = 0.4759907607496632
```

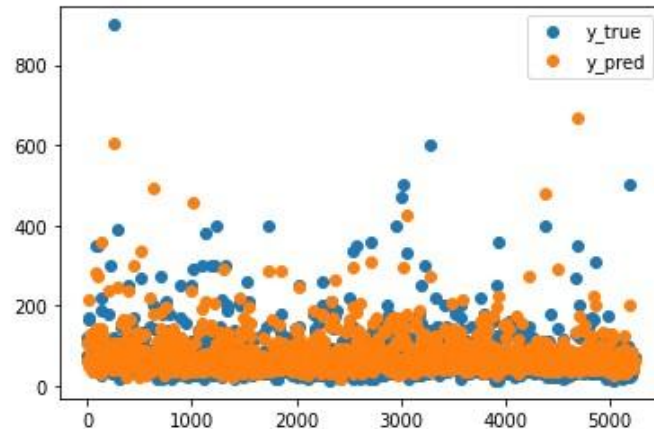
L'écart moyen entre les valeurs réelles et les prédiction est de : 24€



XGB Regressor

```
regressor.score(X_test, y_test) = 0.5569318128121723
```

L'écart moyen entre les valeurs réelles et les prédiction est de : 23€



Démonstration

Deepbnb

Conclusion

Méthode de Travail :

- Distanciel peu adapté → présentiel en A110
- Realtime Notebook → utilisation de Datalore
- Méthode Agile peu adaptée (absence de PO)
→ Répartition des tâches d'une to-do list

Merci pour votre attention !