



ÉCOLE
D'INGÉNIEURS
PARIS-LA DÉFENSE



IMMEST

ESTIMATION IMMOBILIERE A PARTIR DU BIG DATA

Clément BOMPARD – Jérémy CHANSIN – Sory-Bineta DIALLO –
Kimberly HO – Carel MBOTTA-ELIMBI
27/03/2018

Table des matières

Introduction.....	2
Résumé.....	3
IMMEST	4
Récupération des données.....	4
Scrapy	4
Splinter	7
Stockage des données.....	9
Convertir les données csv en sql	9
Exporter les données dans une base de données.....	10
Estimation immobilière	10
Sélection des critères à prendre en compte	10
Location courte durée	10
Location longue durée ou vente	11
Prédiction d'une formule mathématique	11
Visualisation des données	12
Front-end.....	12
Back-end.....	14
Conclusion	16
Annexe 1: Amazon Web Services (AWS)	16

Introduction

IMMEST est un projet unique travaillant sur l'estimation immobilière à l'aide du Big Data à Paris. Ce projet regroupe plusieurs partenaires :

- Bertrand HASSANI – Capgemini,
- Dominique GUEGAN – Labex ReFi (Paris 1 – ENA – ESCP),
- Toutes les ressources de L'ESILV.

De plus, il rassemble cinq étudiants issus de l'Ecole Supérieure Léonard De Vinci (ESILV), avec des spécialisations distinctes :

- Kimberly HO, leader de l'équipe et spécialisé en Informatique,
- Sory-Bineta DIALLO, spécialisé en Informatique,
- Carel MBOTTA-ELIMBI, également spécialisé en Informatique,
- Clément BOMPARD, spécialisé en Ingénierie Financière,
- Jérémy CHANSIN, spécialisé en Ingénierie Financière,

Le mentor d'IMMEST est Gaël CHAREYRON, chef du département informatique, des objets connectés, et du Big Data.

Résumé

Acheter, vendre, ou louer un bien immobilier est devenu une activité courante pour une personne quel que soit les raisons :

- Des étudiants souhaitant étudier ou effectuer un stage (à l'étranger),
- Des touristes cherchant une accommodation complète à louer à bas prix et pour une courte durée,
- Des personnes voulant investir et devenir propriétaire d'un appartement ou encore d'une maison,
- Ou même des professionnels d'agence travaillant sur le marché immobilier et qui aimeraient connaître le prix du marché immobilier actuel,

Nous le savons, il existe plusieurs services en ligne permettant d'estimer le « juste prix » d'un bien immobilier. Ces estimations sont généralement basées sur ces principaux critères :

- Localisation géographique,
- Etat du bien,
- Surface,
- Nombre de pièces (chambres, salon, cuisine, salle de bain),
- Année de construction de l'immeuble,
- Aménagement en plus (terrasse, balcon, cave, ascenseur, belle vue).

C'est ici qu'IMMEST se démarque, nous fournissons un service innovant et opérationnel prenant en considération non seulement les principaux critères, mais également les alentours (restaurants réputés, transports, centre commerciaux, ...). Nous prenons en considération les envies, besoins et habitudes de chaque individu permettant ainsi d'augmenter ou de baisser la valeur du bien immobilier. IMMEST offre une estimation gratuite via une application web ergonomique dédié principalement aux agences immobilières et aux particuliers.

IMMEST

Récupération des données

La première étape de ce projet a été de créer un jeu de données, qui était encore inexistant en début Octobre. Avoir ces données était utiles pour comprendre au mieux le marché de l'immobilier, et de faire des analyses sur les prix des appartements par arrondissement ou quartier.

Pour créer ce jeu de données nous avons eu recours à la méthode du web scraping. Autrement dit une technique d'extraction du contenu de sites Web, via un script ou un programme, dans le but de le transformer pour permettre son utilisation dans un autre contexte, dans notre cas, l'estimation des prix des appartements.

Scrapy

Scrapy est un framework open-source permettant la création de robots d'indexation, c'est à dire un logiciel qui explore automatiquement le Web.

Scrapy a été utilisé pour le site de pap.fr (Figure 1) ainsi que lafourchette (Figure 3). Pour récupérer ces données, un script en python (Figure 4) a été écrit et lancé. Tout simplement, sur ce script nous avons indiqué à Scrapy où trouver les informations utiles, grâce à des balises. En effet pour trouver les balises utiles nous avons dû analyser le code source de l'annonce en question (Figure 2).



The screenshot shows a real estate listing on the pap.fr website. On the left is a photograph of a modern apartment building with a red tennis court in the foreground. To the right of the photo is a star icon in a circle. The listing title is "Location meublée appartement 2 pièces 38 m² Paris 15E" followed by the price "1.400 €" in blue. Below the title is a detailed description: "Paris 15e (75015). Limite 7ème métro Cambronne - Ségur. Très bel appartement de 38 m², légèrement meublé. Dans une résidence sécurisée de grand standing, au milieu de jardins, avec tennis pour les résidents. Au 4ème étage avec..." followed by a link "Lire la suite". Below the description are icons for metro lines M, RER, and T, and the text "Cambronne, Ségur, Sèvres - Lecourbe". At the bottom left of the listing is the reference "Réf. : C07/0624 / 03 mars 2018". At the bottom right are three blue buttons: "2 pièces", "1 chambre", and "38 m²". At the very bottom right is a button with the text "En savoir plus".

Figure 1: Annonce du site pap.fr

```

▼<div class="search-list-item">
  ▼<div class="item-content">
    <div class="infos-box action-annonce hidden" data-annonce="{\"id\":\"420700624\"}">
      </div>
      ▼<a class="item-title" href="/annonces/appartement-paris-15e-r420700624" name="420700624" onclick=
      "ga('send','event','LISTE LOCATION','Consulter','Resume Titre');">
        <span class="h1">Location meublée appartement 2&nbsp;pièces 38&nbsp;m² Paris 15E</span>
        ▼<span class="item-price">
          <strong>1.400&nbsp;€</strong>
        </span>
      </a>
      ▼<p class="item-description" data-gaq="{\"category\":\"LISTE LOCATION\",\"action\":\"Consulter\",\"label\":\"Resume
      Description\"}">
        "
        Paris 15e (75015).
        Limite 7ème métro Cambronne - Ségur. Très bel appartement de 38&nbsp;m², légèrement meublé.

        Dans une résidence sécurisée de grand standing, au milieu de jardins,
        avec tennis pour les résidents.
        Au 4ème étage avec...
        <a href="/annonce/locations-paris-75-g439-r420700624" rel="nofollow">Lire la suite</a>
      </p>
      <p class="item-transports">Cambronne, Ségur, Sèvres - Lecourbe</p>
      <p class="item-date">
        Réf. : C07/0624 / 03 mars 2018
      </p>
    </div class="clearfix">...</div>
  </div>

```

Figure 2: Code source de l'annonce figure 1.

En regardant le code source figure 2, on peut voir dans quelles balises les informations sont écrites. Par exemple :

- Le titre de l'annonce : « Location meublée appartement 2 pièces 38 m² Paris 15^E », se trouve à l'intérieur de la balise *span class="h1"*, qui se trouve en dessous de la balise *a class="item-title"*,
- Le prix de l'appartement : « 1400€ », se situe en dessous de la balise *span* du titre, dans la balise *span class="item-price"*,
- La courte description de l'appartement : « Paris 15e (75015). Limite 7ème métro Cambronne - Ségur. Très bel appartement de 38 m², légèrement meublé. Dans une résidence sécurisée de grand standing, au milieu de jardins, avec tennis pour les résidents. Au 4ème étage avec... », se situe dans la balise *p class="item-description"*,
- Et enfin, les métros près du bien : « Cambronne, Ségur et Sèvres – Lecourbe », se situe dans la balise *p class="item-transports"*.

En général, toutes les annonces du site suivent la même structure, c'est à dire que tous les titres des annonces se trouve dans la balise *span class="h1"*, les prix dans la balise *span class="item-price"*, les descriptions des appartements dans la balise *p class="item-description"* et les métros attenants dans la balise *p class="item-transports"*. Certes, il peut y avoir des annonces qui ne respectent pas cette structure, c'est donc à nous d'analyser et de modifier notre script en fonction de ces annonces.



Figure 3: Annonce du site lafourchette.com

Concernant les restaurants de lafourchette, nous nous intéressons davantage au nom du restaurant, à son adresse et bien sûr sa note. Pour récupérer ces données, nous avons suivi le même principe que pour les annonces immobilières de pap.fr. Nous avons su ainsi, identifier les balises correspondantes et sauvegarder les données.

Scrapy est un framework efficace, mais qui comporte toutefois des failles, dont la première est qu'il n'est pas dans la mesure de récupérer les textes écrits en javascript. Cela nous a posé problème tout particulièrement pour le site de seloger.com. La solution à ce problème a été d'utiliser Splinter.

```
import scrapy

class PapSpider(scrapy.Spider):
    name = "pap"

    def start_requests(self): #donner les url des sites scrapper
        urls = ['https://www.pap.fr/annonce/locations-paris-75-g439']
        for url in urls:
            yield scrapy.Request(url=url, callback=self.parse)# lance request et appelle fonction parse

    def parse(self, response):
        def remove_space(var):
            return ' '.join(var.split())

        for annonce in response.xpath('//div[@class="box search-results-item"]'):
            title = ' '.join(annonce.xpath('//span[@class="h1"]//text()').extract())
            description = ' '.join(annonce.xpath('//p[@class="item-description"]//text()').extract())
            transport = ' '.join(annonce.xpath('//p[@class="item-transports"]//text()').extract())
            price = ' '.join(annonce.xpath('//span[@class="price"]//text()').extract())
            room = ' '.join(annonce.xpath('//ul[@class="item-summary float-left"]//text()').extract())

            yield {
                "title": remove_space(title),
                "description": remove_space(description),
                "transport": remove_space(transport),
                "price": remove_space(price),
                "room": remove_space(room),
            }

            page_details = annonce.xpath('//div[@class="float-right"]/a/@href').extract_first()
            description_details = ' '.join(annonce.xpath('//p[@class="item-description"]//text()').extract())
            yield {
                "description_details": remove_space(description_details),
            }

        next_page = response.xpath('//li[@class="next"]/a/@href').extract_first()
        print(next_page)

        if next_page is not None:
            next_page = response.urljoin(next_page)
            yield scrapy.Request(next_page, callback=self.parse)
```


Figure 4: Script Scrapy.

Splinter

Splinter, est tout comme Scrapy, un outil open source permettant la création de robot. A la différence de Scrapy, Splinter permet d'automatiser diverses actions sur un navigateur (ex : Mozilla Firefox, Google Chrome, ...), tel qu'aller sur un moteur de recherche quelconque (ex : google, bing, duck duck go, ...) et interagir avec les sites, en cliquant sur des boutons, en visitant des urls etc Concrètement Splinter agit comme un humain.

Tout comme Scrapy, nous avons écrit un script en python permettant d'aller sur le site de seloger.com (Figure 6 et 7), cliquer sur chaque annonce, et récupérer les informations utiles (Figure 5) :

- L'avis du professionnel,
- Les caractéristiques générales de l'appartement,
- Et les caractéristiques propres à l'appartement.



Appartement

2 pièces 1 chambre 38 m²

Paris 18ème

1 085 €^{CC}

Le bien
L'agence
Le quartier
Votre I

L'AVIS DU PROFESSIONNEL

RUE ORDENER, au niveau de la rue Damrémont, appartement de 2 pièces repeint à neuf. D'une surface de 38m², il se compose d'une entrée, un séjour, une chambre, une cuisine, une salle de bains, un WC séparé. 4ème étage sans ascenseur. Chauffage individuel au gaz. Honoraires loi Alur à la charge du locataire = 570 euros.

GÉNÉRAL

Surface de 38 m ²	Année de construction 1900	Au 4ème étage	2 Pièces	1 Chambre
------------------------------	----------------------------	---------------	----------	-----------

A L'INTÉRIEUR

1 Salle de bain	1 Toilette	Toilettes Séparées	Chauffage central gaz	Entrée
-----------------	------------	--------------------	-----------------------	--------

Salle de Séjour

Figure 5: Extrait d'annonce du site seloger.com


```

from splinter import Browser
import time
import json

browser = Browser()
browser.visit('http://www.seloger.com/list.htm?tri=initial&idtypebien=2,1&idtt=1&ci=750102&naturebien=1,2,4')
#PARIS 1er : http://www.seloger.com/list.htm?tri=initial&idtypebien=2,1&idtt=1&ci=750101&naturebien=1,2,4

##### PARTIE 1 #####

data = []
file = open("url_annonces_locations_paris2.txt", "w")
while True:
    liste_resultat = browser.find_by_xpath('//section[@class="liste_resultat"]')
    res1 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-sl c-pa-gold cartouche "]')
    res2 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-bd c-pa-gold cartouche "]')
    res3 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-sl c-pa-silver cartouche "]')
    res4 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-bd c-pa-silver cartouche "]')
    res5 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-sl cartouche "]')
    res6 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-bd cartouche "]')
    res7 = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-lv cartouche "]')
    res = res1+res2+res3+res4+res5+res6+res7
    print(len(res))
    for ele in res:
        href = ele.find_by_tag('a')
        link = href['href']
        data.append(link)
        file.write(str(link) + "\n")
        # print(link)
    next_page = browser.find_by_xpath('.//a[@class="pagination-next"]')
    link_nextpage = next_page['href']
    print(link_nextpage)
    browser.visit(link_nextpage)

print(data)
file.close()

```

Figure 6: Script Splinter Partie 1(Récupération des urls de chaque annonce).

```

##### PARTIE 2 #####
file = open("url_annonces_locations_paris2.txt", "r")
data=[]
for line in file:
    data.append(line)
file.close()

liste_resultat = browser.find_by_xpath('//section[@class="liste_resultat"]')
res = liste_resultat.find_by_xpath('//div[@class="c-pa-list c-pa-sl c-pa-gold cartouche "]')

##### STORE VALUES INTO JSON #####
with open('seloger_paris2.json', 'w') as f:

    for i in range(len(data)):
        browser.visit(data[i])
        selogerdata={}
        title = browser.find_by_tag('h1').text
        description = browser.find_by_id('js-descriptifBien').text
        price = browser.find_by_id('price').text
        # print(title)
        # print(description)
        selogerdata["title"] = title+"\n"
        selogerdata["description"] = description+"\n"
        selogerdata["price"] = price+ "\n"

        cat = browser.find_by_xpath('//section[@class="categorie"]')
        for ele2 in cat:
            cat_list = ele2.find_by_xpath('.//div[@class="g-col g-20"]')
            # print(len(cat_list))
            for ele3 in cat_list:
                mots = ele3.find_by_xpath('.//div[@class="u-left"]').text
                # print(mots)
                mots_data = []
                mots_data.append(mots)
                for j in mots_data:
                    selogerdata["categorie"+j] = mots+"\n"

    print(selogerdata)
    json.dump(selogerdata,f)
    f.write("\n")

```

Figure 7: Script Splinter Partie 2 (Récupération des détails de chaque annonce).

Stockage des données

Convertir les données csv en sql

Les données récupérées via la méthode du scraping ont été soit au format csv (Comma-separated values) autrement dit excel, mais aussi au format json (JavaScript Object Notation) (Figure 8) :

```

"categorieEntr\u00e9e": "Entr\u00e9e\n",
"description": "Quai du Louvre: Au 2\u0008me \u0009tage donnant sur la Seine, charmant 2 pi\u0008sces comprenant une entr\u00e9e, un s\u0009jour / salle \u0009p\u0009d\u0009t de garantie: 4780 euros Honoraires de location: 2390 euros TTC.\n",
"categorieAu 2\u0008me \u0009tage": "Au 2\u0008me \u0009tage\n",
"price": "2 490 \u02ac CC\n",
"title": "Location Appartement 54,8m\u00b2 Paris 1er - Saint Germain l'Auxerrois\n",
"categorieSurface de 54,8 m\u00b2": "Surface de 54,8 m\u00b2\n",
"categorieSalle de bain": "1 Salle de bain\n",
"categorieRangements": "Rangements\n",
"categorieToilette": "1 Toilette\n",
"categorieChambre": "1 Chambre\n",
"categorieChauffage individuel \u0009lectrique": "Chauffage individuel \u0009lectrique\n",
"categorieMeubl\u0009": "Meubl\u0009\n",
"categorie2 Pi\u0008sces": "2 Pi\u0008sces\n",
"categorieSalle de S\u0009jour": "Salle de S\u0009jour\n",
"categorieCuisine s\u0009par\u0009e": "Cuisine s\u0009par\u0009e\n",
"categorieSalle \u0009 Manger": "Salle \u0009 Manger\n"
}
}
"description": "Rue du Roule PENTHOUSE de 126,82 m\u00b2 LC - 130,43 m\u00b2 au sol au 5e \u0009tage \u0009tage d'un tr\u0008s bel immeuble r\u0009habilit\u0009 chambre principale sur cour avec dressing et salle de bains (douche / baignoire) / WC, d'une chambre tr\u0008s lumineuse sur rue avec salle de bains et un l'entretien de l'aquarium et de la terrasse ainsi qu'un acc\u0008s internet hors VOD et appels payants. BAIL SOCI\u0009T\u0009 / R\u0009SIDENCE SECONDAIRE\n",
"categorieCuisine am\u0009ricaine": "Cuisine am\u0009ricaine\n",
"title": "Location Appartement 126,82m\u00b2 Paris 1er - Les Halles\n",
"categorieSurface de 126,82 m\u00b2": "Surface de 126,82 m\u00b2\n",
"price": "5 800 \u02ac CC\n",
"categorie3 Pi\u0008sces": "3 Pi\u0008sces\n",
"categorieTerrasse": "1 Terrasse\n",
"categorieAu 5\u0008me \u0009tage": "Au 5\u0008me \u0009tage\n",
"categorie2 Chambres": "2 Chambres\n",
"categorieB\u0009timent de 6 \u0009tages": "B\u0009timent de 6 \u0009tages\n",
"categorieSalle de S\u0009jour": "Salle de S\u0009jour\n"
}
}

```

Figure 8: Exemple de données en format json.

Pour les insérer dans une base de données SQL, nous avons converti toutes ces données en format SQL (Figure 9) via un convertisseur automatique en ligne :

```
CREATE TABLE mytable(
  Idlafourchette INTEGER NOT NULL PRIMARY KEY
  ,title          VARCHAR(63) NOT NULL
  ,address        VARCHAR(89) NOT NULL
  ,Averageprice   INTEGER
  ,rating         VARCHAR(3)
);

INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (1,'K- Moon','17 Boulevard de Belleville 75011 Paris','16','10');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (2,'Bistrot Wattignies','1 Rue de Wattignies 75012 Paris','27','9,6');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (3,'La Grande Cascade - Frédéric Robert','Bois de Boulogne, allée de Longchamp 75016 Paris','150','9,5');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (4,'Mandaloune','90 Rue didot 75014 Paris','15','9,4');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (5,'Upper Café- Les Halles','66 Rue des Lombards 75001 Paris','17','9,5');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (6,'Le Poulpry','12, rue de Poitiers 75007 Paris','50','9,3');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (7,'Aarchna','19, rue du Point-à-Graphie 75028 Paris','28','9,1');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (8,'Carte Blanche','88 Bis Avenue Kléber 75116 Paris','60','9,1');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (9,'Maison Rostang','20 Rue Rennequin 75017 Paris','175','9,6');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (10,'Ao Izakaya','12 Rue de Caumartin 75009 Paris','45','9,5');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (11,'Joeric','47 Rue des 3 Bornes 75011 Paris','25','9,3');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (12,'Les Rupins','35 Boulevard de Magenta 75010 Paris','20','9,2');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (13,'Mayfair Garden','26 Rue Lauriston 75116 Paris','15','9,2');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (14,'Chez Petit Hugo','102Bis Avenue du Général Michel Bizot 75012 Paris','15','9,3');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (15,'Le Baudelaire - Hôtel Le Burgundy Paris','6/8, rue Duphot 75001 Paris','80','9,6');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (16,'L'Abeylle - Shangri-La Hotel, Paris','10, avenue d'Iena 75016 Paris','220','9,7');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (17,'La Scène Théâtre','18 Rue Troyon 75017 Paris','125','9,4');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (18,'Addict Factory','30 Boulevard de Vaugrard 75015 Paris','18','9,2');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (19,'Homies del Mar','13 Rue Beaurellillis 75004 Paris','35','9,3');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (20,'Autour du Saumon Miron','64, rue François Miron 75004 Paris','42','9,2');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (21,'Sangeet','64 Rue de la Glacière 75013 Paris','18','9');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (22,'1 place Vendôme - Hôtel de Vendôme','1, place Vendôme 75001 Paris','70','9,3');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (23,'O Maquis','119 Rue de Tocqueville 75017 Paris','33','9,4');
INSERT INTO mytable(Idlafourchette,title,address,Averageprice,rating) VALUES (24,'Le 23 Clauzel - Julie Rivière -','73, rue Clauzel 75009 Paris','50','9,1');
```

Figure 9: Exemple de données en format sql.

Exporter les données dans une base de données

Ainsi, une fois que toutes nos données étaient en format SQL, nous avons pu facilement les insérer dans notre base de données MySQL.

Nous avons tout d'abord stocké nos données en locale, sur nos propres machines, puis grâce à notre partenaire, nous avons pu ouvrir un cloud pour les stockés, car elles sont conséquentes. Nous avons pu alors créer un compte sur Amazon Web Services (AWS) (voir Annexe 1). Malheureusement, nous avons des problèmes pour se connecter au server, et nous ne l'avons donc pas utilisé.

Estimation immobilière

Sélection des critères à prendre en compte

La première étape afin de pouvoir déterminer le prix d'un logement est de choisir les différents critères à prendre en compte, et lesquels sont les plus importants pour déterminer un prix, que ce soit pour une vente ou une location courte durée. Il y a évidemment les critères de bases du logement qui rentrent en compte, tel que le nombre de chambres, la surface, ou encore le type de chauffage. Puis il y a aussi le quartier et l'environnement extérieur à prendre en compte. Ces éléments extérieurs au logement peuvent faire baisser ou augmenter le prix. Par exemple, être proche des lignes de transports augmentera le prix alors qu'au contraire, un logement à l'écart des lignes de transports verra son prix diminué.

Afin de déterminer tous ces critères, qu'ils soient propres ou non à l'appartement, nous avons réfléchi à ce qui pourrait faire varier le prix. Nous sommes aussi allés voir dans des agences afin d'obtenir plus d'informations, étant donné que nous avons peu de connaissance en matière d'immobilier. Nous avons obtenu peu d'informations, mis à part la confirmation qu'il n'existe pas de formules mathématiques pour estimer le prix d'un bien. Nous avons donc dû effectuer de nombreuses recherches afin de déterminer quels critères sont utiles ou non pour estimer.

Nous en sommes arrivés à la conclusion que les critères n'étaient pas les mêmes pour une vente ou une location courte durée (AirBnB par exemple). Pour une vente, les critères propres à l'appartement sont plus importants que pour une location courte durée, pour qui la proximité aux monuments et transports, mais aussi la localisation dans un quartier vivant est plus appréciée.

Location courte durée

Concernant la location courte durée (exemple Airbnb), nous avons concentrés nos critères sur les besoins des locataires. Par exemple, Il était primordial d'apporter un pourcentage plus important sur le critère des activités quotidienne des locataires (monuments, centre commerciaux, restaurants, ...). Car il faut savoir que plusieurs locataires choisissent leur logement en fonction des activités qu'ils aimeraient faire lors de leur séjour, que ce soit touristique, professionnel, familial ou même pour une affaire de santé. Pour ces raisons-ci, nous avons dû étudier et prendre en compte chaque potentiel activité à proximité de chaque logement afin d'adapter un prix en fonction des différents critères à proximité.

Location longue durée ou vente

Pour la location longue durée, les critères qui prédominent sont ceux concernant directement le logement, mais aussi les transports qui sont primordiaux dans la vie quotidienne. Par exemple, posséder des toilettes séparées est un plus pour une vente, alors que cela a un impact moindre sur une location courte durée. Au contraire d'une location courte durée, les activités quotidiennes tels que les restaurants ou encore les centres commerciaux ont une importance moindre.

Prédiction d'une formule mathématique

Après avoir effectué des recherches, nous nous sommes rendu compte qu'il n'existait pas de formules mathématiques pour prédire le prix d'un appartement. C'est pourquoi nous avons décidé de créer notre propre formule. Cette dernière est basée sur la « puissance » des différents critères mais aussi sur le prix moyen du mètre carré selon l'arrondissement.

Tout d'abord, il faut multiplier le prix moyen au mètre carré par la superficie de l'appartement, puis selon les différents critères, ajouter ou enlever de la valeur grâce aux différents pourcentages appliqués à chaque critère. Prenons un exemple concret :

Un appartement de 85 m², situé dans le 12^{ème} arrondissement. On a donc :

$$85 \times 8130 \text{ (prix moyen au m}^2 \text{ dans le 12e Arrondissement)} = 691050 \text{ Euros}$$

A partir de ce chiffre, on ajoute des pourcentages. Par exemple, chaque m² d'un balcon faut hausser le prix de 0,5%. Pour un balcon de 15m², le prix augmentera de 7,5%. Il faudra donc ajouter à 691050 : $691050 \times 7,50\% = 51828$. Il faut répéter cette méthode pour tous les autres critères (certains critères font aussi baisser le prix tel qu'un faible ensoleillement ou encore des nuisances sonores régulières).

Au final, la répartition du prix se fera de telle manière pour cet appartement (Figure 10):

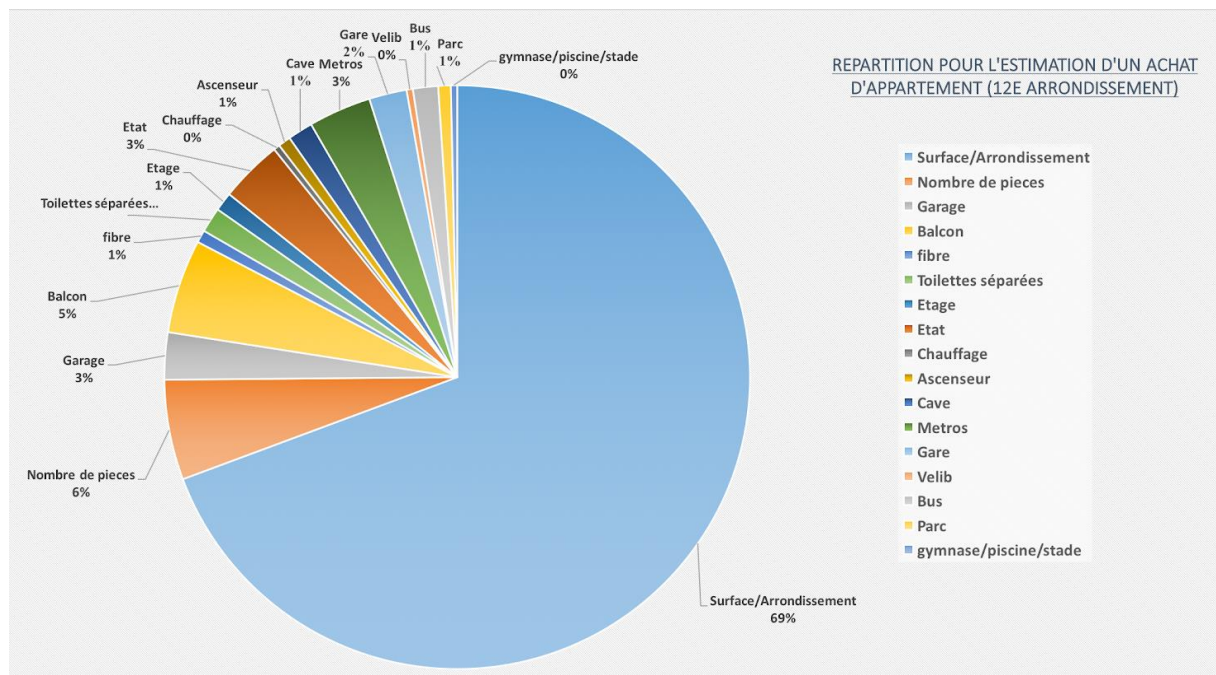


Figure 10: Répartition du prix pour un achat d'appartement

Visualisation des données

Pour la partie visualisation, l'objectif était de faire une application permettant de donner une estimation des biens immobiliers qu'auraient renseignés l'utilisateur.

Une carte qui situerait le bien que l'utilisateur voudrait estimer avec tout autour les commerces musées et transports les plus proches afin de percevoir en quoi ces critères modifient le prix du bien.

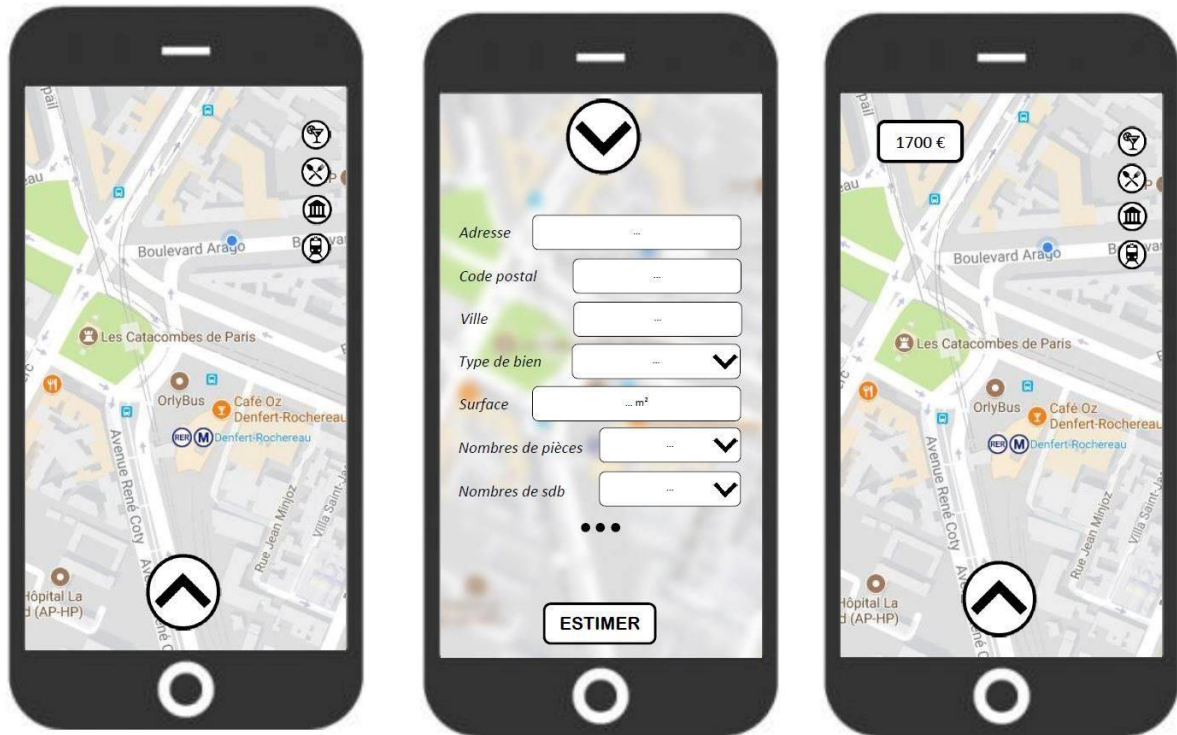


Figure 11: Prototype de l'application mobile

Pour un souci de portabilité, c'est-à-dire que l'application puissent être utilisables par toutes les plateformes, le choix du langage de développement s'est porté sur le JavaScript. Donc se lancer dans la création d'une web application, disponible sur tous les systèmes d'exploitation et sur tous les appareils que ce soit ordinateurs, tablettes ou smartphones.

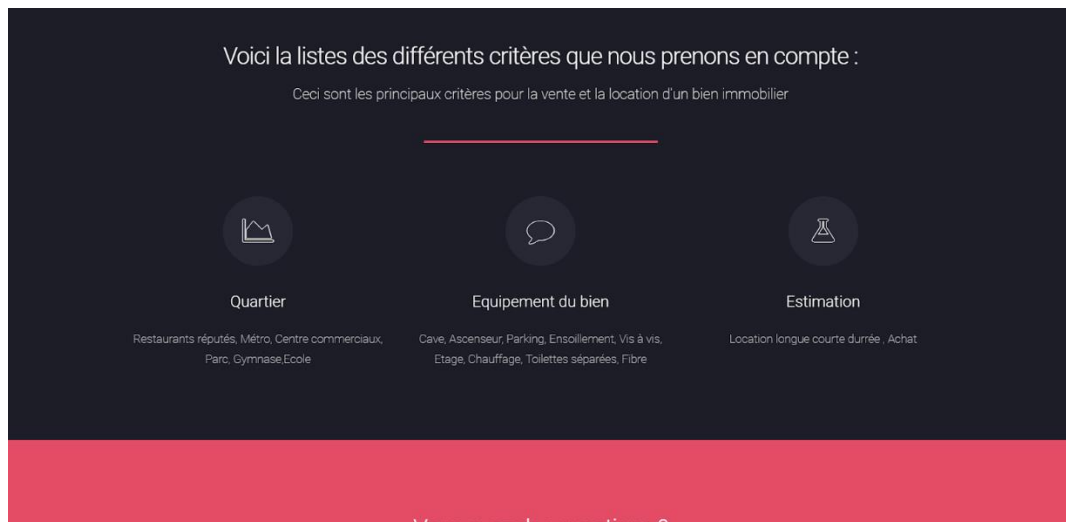
Ainsi, après quelques réarrangements sur la partie design, l'application web a été mise en place.

Front-end

Le formulaire pouvant être rempli par un utilisateur afin d'évaluer le bien à estimer.



(a)



(b)

(c)

Figure 12 (a) (b) (c): Page d'accueil et formulaire.

Pour le Front-End de l'application, les technologies utilisées sont :

- Le JavaScript qui permet de dynamiser le site web, et utile pour les formulaires
- Le HTML et le CSS pour la beauté du site et son affichage
- La carte quand elle, est chargée de l'Api GoogleMap et on en utilise les fonctionnalités afin de charger la carte avec les appartements. A ce fait, on utilise du PHP qui permet au serveur d'interroger la base donnée avec toutes les données des appartements récupérées.

Il a été nécessaire d'obtenir une Google Map Key afin de pouvoir utiliser ce module.

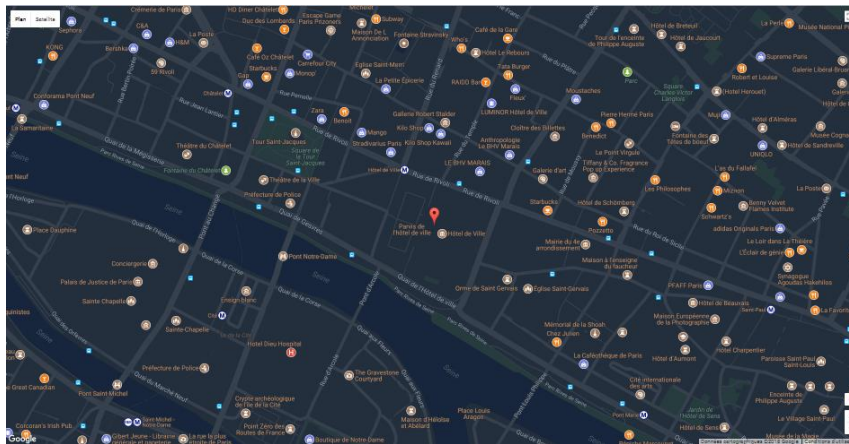


Figure 13: Carte.

Back-end

- Insertions des données récupérées dans les bases de données Mysql
- Réorganisation de ces données dans le fichier excel.

Les technos utilisées pour communiquer avec la base de données et alimenter la carte nous ont été fournies par les fonctionnalités de l'Api Google Map.

```
function addMarker(address, info){
    var geocoder = new google.maps.Geocoder();
    geocoder.geocode( { 'address': address}, function( data, status) {
        if( status === google.maps.GeocoderStatus.OK){
            var pt = data[0].geometry.location;
            bounds.extend(pt);
            var marker = new google.maps.Marker({
                position: pt,
                icon: icon,
                map: map
            });
            var popup = new google.maps.InfoWindow({
                content: info,
                maxWidth: 300
            });
            google.maps.event.addListener(marker, "click", function() {
                if (currentPopup != null) {
                    currentPopup.close();
                    currentPopup = null;
                }
                popup.open(map, marker);
                currentPopup = popup;
            });
            google.maps.event.addListener(popup, "closeclick", function() {
                map.panTo(center);
                currentPopup = null;
            });
        } else {
            alert("Geocode was not successful for the following reason: " + status);
        }
    })
}
```

Figure 14: Script.

Donc la fonction `addMarker` permet d'ajouter des marqueurs contenant les appartements scrapés et sauvegardés dans une base de données.

Tout d'abord en convertissant l'adresse renseignées dans la base de données avec la méthode `géocode` :

- “`géocode`” permet de convertir l'adresse en point coordonnées, latitude longitudes,
- “`google.maps.Marker`” permet de créer un objet `marker` qui s'affiche sur la `googleMap`,
- “`google.maps.InfoWindow`” permettant de créer une fenêtre `popup` d'information pour les marqueurs ainsi, afficher l'estimation faite.

Conclusion

À travers ce projet, nous avons pu voir les différentes facettes d'un marché, le marché de l'immobilier. Nous n'avions aucunes connaissances du marché immobilier, nous avons nous même dû apprendre les choses essentielles à propos de ce secteur et ce projet nous a permis de comprendre son fonctionnement : comment sont définis les prix, ce qui les impactent ou encore sur les différents types d'investissement possibles. Nous avons donc pu voir ces différents points grâce à notre travail, au travers des différentes contraintes liées au monde de l'immobilier.

Ce projet nous a aussi permis d'apprendre par nous-mêmes de nouvelles techniques telles que le scraping, mais surtout à être autonome, à choisir nous-mêmes sur quel support nous voulons travailler.

Pour finir, ce travail en groupe à permis à chaque membre d'apprendre à mieux travailler en équipe, avec les points forts et faibles de chacun, pour mener à bien ce projet, ce qui est un point essentiel pour la suite de nos carrières respectives.

Annexe 1: Amazon Web Services (AWS)

Amazon Web Services (AWS) est une plateforme de services de cloud sécurisée, qui met notamment à disposition de la puissance de calcul et des fonctionnalités de stockage de base de données.

Ce service nous a été fourni par notre partenaire Bertrand Hassani (Capgemini). Nous avons ouvert notre proposer server qui nous permettrait de stocker nos données, faire des analyses mais aussi développer notre application.

Amazon Elastic Compute Cloud (Amazon EC2) est un service Web qui fournit une capacité de calcul sécurisée et redimensionnable dans le cloud. Destiné aux développeurs, il est conçu pour faciliter l'accès aux ressources de cloud computing à l'échelle du Web.

Sur Amazon EC2, nous avons choisi plusieurs instances qui satisfaisait nos besoins :

- Modèle : m4xlarge
- vCPU : 4
- Mémoire : 16 (Gio)
- Stockage sur SSD: EBS uniquement (Go)
- Bande passante dédiée à EBS (Mbit/s) : 750

M4

Les instances de type M4 offrent un ensemble équilibré de ressources de calcul, de mémoire et de réseau. Il s'agit d'une option adaptée à de nombreuses applications.

Caractéristiques :

- Processeurs 2,3 GHz Intel Xeon® E5-2686 v4 (Broadwell) s ou 2,4 GHz Intel Xeon® E5-2676 v3 (Haswell)
- Optimisées pour EBS par défaut, sans frais supplémentaires
- Prise en charge de la mise en réseau améliorée
- Equilibre entre les ressources de calcul, de mémoire et de réseau

Modèle	vCPU	Mém. (Gio)	Stockage sur SSD (Go)	Bande passante dédiée à EBS (Mbit/s)
m4.large	2	8	EBS uniquement	450
m4.xlarge	4	16	EBS uniquement	750
m4.2xlarge	8	32	EBS uniquement	1 000
m4.4xlarge	16	64	EBS uniquement	2 000
m4.10xlarge	40	160	EBS uniquement	4 000
m4.16xlarge	64	256	EBS uniquement	10 000

Figure 15: Instance Amazon EC2.

Etant donné que nous avons récupérer beaucoup de données, nous avons privilégier un espace mémoire suffisant, donc 16 Gio, et une bonne bande passante (qui représente le volume de données pouvant être transporté d'un point à un autre dans un laps de temps donné), 750 Mbit/s.

Pour se connecter au server via notre machine, nous devons utiliser :

- Une clé,
- Et le (Domain Name System) DNS.

Le DNS permet entre autres de traduire un nom de domaine en adresse IP afin que nos requêtes puissent arriver jusqu'au serveur cible.

Nous avons testé la connexion à partir d'une machine Windows mais aussi à partir d'une machine Linux, mais sans succès nous avons pu nous connecter au server.