Github

ZILLOW

Zillow a été lancé en 2006 et son siège social est à Seattle. Elle est le premier marché immobilier et locatif dédié à l'autonomisation des consommateurs grâce aux données.

Elle permet à ses clients de profiter de l'inspiration et des connaissances de plus de 110 millions de maison disponible sur leurs base de données et de les connecter avec les meilleurs professionnels locaux afin de les aider dans leur travaux et autre aménagement.

Zillow dessert le cycle de vie du marché immobilier complet : achat, vente, location, financement, rénovation et plus encore. La base de données de Zillow grosse de plus de 110 Millions de maison regroupe toute les catégories aussi bien la location que l'achat avec également des exclusivité non disponible sur le marché Mais ce qui a fait leur renommée est que Zillow exploite la suite la plus populaire d'applications immobilières mobiles, avec plus de deux douzaines d'applications sur toutes les plateformes majeures.

La principale nouveauté de cette plateforme et ce qui en a fait la renommé est Zestimate, un algorithme développé au sein de l'entreprise et qui permet de prédire le plus d'achat de vente ou même de location (avec Rent Zestimate). Cette algorithme étant très précis, il a permis à des millions de personnes de se lancer dans l'achat ou la vente de leur maison de manière confiante.

Récemment Zillow a lancer, sur la plateforme kaggle, un concours afin d'améliorer encore plus leur algorithmes, a la clef plus d'un million de dollars US. En effet cette algorithmes vaut son pesant d'or car ils leur permettrait de réaliser encore plus de profit et d'asseoir leur position de leader sur le marché immobilier

Le But

Zillow a donc déployer sur kaggle leur challenge, mais n'on pour autant pas dévoiler leur algorithmes. Ils ont au contraire dévoilé le logarithmes de la différence entre leur Zestimate et le prix réel de vente des propriétés sur 2 période une de 2016 et l'autre de 2017.
Le but de cette exercice est donc de prédire ce logarithme erreur.

Github

Les Données

Zillow nous as mis a disposition plus de 2 500 000 de maisons et autre local immobilier en vente sur la période de 2017. Chaque ligne est définie par 58 variables :

- airconditioningtypeid : type d'air conditionné (si présent)
- architecturalstyletypeid : style architectural
- basementsqft : surface habitable en sous sol
- bathroomcnt : nombre de salle de bains dont salle d'eau
- bedroomcnt : nombre de chambre
- buildingqualitytypeid : niveau de délabrement
- buildingclasstypeid : type de matériau de soutien
- calculatedbathnbr : nombre de salles de bain sans salles d'eau
- decktypeid : type de porche (si présent)
- threequarterbathnbr : nombre de salles d'eau dans la maison
- finishedfloor1squarefeet : surface habitable au premier étage
- calculatedfinishedsquarefeet : surface habitable totale
- finishedsquarefeet6 : surface habitable finie et non finie
- finishedsquarefeet12 : surface habitable finie
- finishedsquarefeet13 : périmètre de la surface
- finishedsquarefeet15 : surface total
- finishedsquarefeet50 : taille de la surface habitable finie au premier etage
- fips : Federal Information Processing Standard code
- fireplacecnt : nombre de cheminée
- fireplaceflag : présence de cheminée
- fullbathcnt : nombre de salle de bains complète
- garagecarcnt : nombre de garage (incluant les garages rattachés à la maison)
- garagetotalsqft : surface total en pieds carré des garages
- hashottuborspa : presence de jacuzzi ou de spa
- heatingorsystemtypeid : type de chauffage
- latitude : géolocalisation
- longitude : géolocalisation
- lotsizesquarefeet : surface total du terrain en pieds carre
- numberofstories : nombres d'étages
- parcelid : identifiant unique
- poolcnt : nombre de piscine
- poolsizesum : taille totale des piscines (en pieds cubes)
- pooltypeid10 : spa ou jacuzzi
- pooltypeid2 : piscine avec spa ou jacuzzi
- pooltypeid7 : piscine sans jacuzzi ou spa
- propertycountylandusecode : numéro de la localité
- propertylandusetypeid : type d'utilisation du terrain
- propertyzoningdesc : autorisation type d'utilisation de la propriété
- rawcensustractandblock : id du recensement
- censustractandblock : id du recensement
- regionidcounty : id du comté
- regionidcity : id de la ville

Github

- regionidzip : code postal de la ville
- regionidneighborhood : id du quartier
- roomcnt : nombre total de pièces
- storytypeid : type d'étage
- typeconstructiontypeid : type des matériaux de construction
- unitcnt : nombre d'unité de la structure (duplex,triplex ..)
- yardbuildingsqft17 : patio dans la cours
- yardbuildingsqft26 : cabane de stockage
- yearbuilt : année de construction
- taxvaluedollarcnt : valeur totale imposable de la parcelle
- structuretaxvaluedollarcnt : valeur imposable de la structure bâtie
- landtaxvaluedollarcnt : valeur imposable du terrain
- taxamount : taxe foncière totale pour cette année
- assessmentyear : année de l'évaluation de l'impôt foncier
- taxdelinquencyflag : taxes foncières pour cette parcelle impayée depuis 2015
- taxdelinquencyyear : Année pour laquelle les taxes foncières impayées étaient dues

Groupage des données

Les données ont ainsi pu etre repartis en plusieurs groupe, premièremetn les donnees continues numérique, ensuite les donnees categorique qui sont utilisé pour decrire des type, enfin les données booleenne qui servent ici a signaler la presence d'une autre colonne.

Données numérique
basementsqft, bathroomcnt, bedroomcnt, calculatedbathnbr, threequarterbathnbr, finishedfloor1squarefeet, finishedsquarefeet6, finishedsquarefeet12, finishedsquarefeet13, finishedsquarefeet15, finishedsquarefeet50, fireplacecnt, fullbathcnt, garagecarcnt, garagetotalsqft, latitude, longitude, lotsizesquarefeet, numberofstories, poolcnt, poolsizesum, roomcnt, unitcnt, yardbuildingsqft17, yardbuildingsqft26, yearbuilt, taxvaluedollarcnt, structuretaxvaluedollarcnt, landtaxvaluedollarcnt, taxamount, assessmentyear, taxdelinquencyyear

Données d'identification(Id)
airconditioningtypeid, architecturalstyletypeid, buildingqualitytypeid, buildingclasstypeid, decktypeid, fips, heatingorsystemtypeid, parcelid, pooltypeid10, pooltypeid2, pooltypeid7, propertycountylandusecode, propertylandusetypeid, propertyzoningdesc, rawcensustractandblock, censustractandblock, regionidcounty, regionidcity, regionidzip, regionidneighborhood, storytypeid, typeconstructiontypeid

Flag
fireplaceflag, taxdelinquencyflag

Github

Choix des données

Pour sélectionner les données que nous allions effectivement utiliser pour obtenir un modèle de régression linéaire nous avons d'abord évalué chacunes des colonnes pour déterminer si elles étaient suffisamment représentée dans le jeu de données et si elle n'était pas redondante avec une autre colonnes pour éviter de donner trop d'importance à un trait n'ayant rien de particulier par rapport aux autres.

Sur le remplissage des données manquante

De manière classique les données numérique continue manquante ont été remplacé par la moyenne de cette données, ou bien par la médiane, et ceux afin de ne pas fausser la répartition. Cependant il est important de distinguer d'autre colonne ou l'absence de données veut en fait signifie 0, comme par exemple la surface de piscine ou si la données n'est pas remplis cela correspond a 0 car il n'y a pas de piscine sur le terrain

Les données catégoriques ont elles été remplis de deux manière. Dans ce jeu de données il arrive que l'absence de remplissage correspond à l'absence totale du caractères comme le type de porche par exemple a ce moment la on peut le remplir par 0 pour signifier l'absence. Pour le reste des données catégorique, deux méthodes ont été utilisé premièrement les remplirs aléatoirement en conservant la répartition statistique de cette donnée cela permet de ne pas fausser cette colonne, mais augmente le bruit de la ligne, deuxièmement prédire grâce à un random forest la valeur qui devrait être renseignée, méthode qui introduit du bruit dans la colonne, mais conserve l'intégrité de la ligne. Il serait également intéressant de remplir les id de location (county, zipcode, ...) par le résultat d'une recherche réalisée à partir de la latitude et la longitude.

Les flags sont seulement des colonnes indiquant la présence d'une autre colonne, il faut donc les retirer car le remplissage des données catégorique nous a permis de fusionner ces informations directement dans les données catégoriques.

Idée de Feature Engineering

Une des plus grande importance de l'achat et de vente immobilier reste la localisation. Parmis nos idées de Feature Engineering, la plus importante serait donc de déterminer, grâce à la longitude et la latitude, la proximité du lieu a des endroits comme la plage (car nos data sont sur les rebords de los Angeles), centre commerciaux ou encore les transports en commun. Ces nouvelles données apporterait énormément d'information quant au prix pratiqué.

Github

Résultats

Train 2017 - Test 2016

rootMeanSquaredError	0.16093483082776017
meanAbsoluteError	0.06926029651404056
explainedVariance	1.4942586929040245E-4
meanSquaredError	0.025900019773559782
r2	0.0017761439267559576

Train 2016 - Test 2017

rootMeanSquaredError	0.17072574812691294
meanAbsoluteError	0.07009107756725128
explainedVariance	1.2654433988285878E-4
meanSquaredError	0.029147281073494114
r2	0.0018770834087459276

Problemo Prompto

Même après allocation de 15 go de ram et configuration du garbage collector, le programme servant à la prédiction des ids manquante par arbre de décision continue de nous renvoyer des erreurs quant à la mémoire. Les résultats dont nous discuterons ci après ne peuvent donc pas être établis par rapport au données remplis par cette méthode.