Meet-up ML: Énoncé de la problématique

Détection de toits de bâtiments à partir d'images satellites

Intact Assurance et Co-operators

10 novembre 2018

Mise en contexte

L'assurance des entreprises consiste essentiellement à assurer les biens, la perte d'exploitation ou la responsabilité civile d'une entreprise. Afin d'offrir un service de pointe, la compagnie d'assurance **Assure-Toit** désire avoir un système qui permet d'estimer rapidement la prime à charger à une entreprise en posant le moins de questions possibles au client. Une question régulièrement demandée est si le bâtiment commercial possède un toit vert (*green roof*). Cette particularité change légèrement le risque en lien avec le bâtiment, ce qui entraîne une légère hausse de prime.

Afin d'éviter de poser systématiquement la question au client, la compagnie a suggéré utiliser l'adresse du bâtiment pour extraire une image satellite du toit. Ensuite, elle prévoit utiliser un modèle permettant de traiter ces images et ainsi reconnaître automatiquement les bâtiments ayant des toits verts.

Votre tâche consiste donc à bâtir un modèle permettant de faire une classification binaire d'images satellites. Étant donné qu'il est difficile d'obtenir une image précise et centrée à partir d'une adresse, le modèle doit seulement reconnaître s'il y a au moins un toit vert sur l'image. De cette façon, l'agent de la compagnie pourra poser la question au client seulement dans le cas où au moins un toit vert est détecté par le modèle.

Objectif

L'objectif est de faire une classification binaire permettant de reconnaître les images satellites ayant au moins un toit vert sur l'image. Vous devrez bâtir un modèle qui fait la classification et présenter votre méthodologie de manière vulgarisée (voir les critères d'évaluation plus bas).

Jeu de données

Le jeu de données qui vous ai proposé provient de données ouvertes de la ville de Chicago. Les images satellites ont été extraites à partir des coordonnées *GPS* rattachées aux permits de construction accordés à Chicago depuis 2006.

Pour entraîner votre modèle, vous pourrez utiliser deux approches différentes. La première consiste à entraîner un modèle directement à partir des images satellites. Ces images sont regroupées dans le fichier *data-train.zip*. Chaque image possède un identifiant (ID) dans le nom du fichier (*image-ID.png*). Cet identifiant vous permet de savoir quelles images possèdent au moins un toit vert, et lesquelles n'en possèdent aucun. L'étiquette pour chacune des images se trouve dans le jeu de données *data-id-train.csv*.

La deuxième approche consiste à utiliser des variables que nous avons déjà trouvées pour vous. Ces variables ont été trouvées en utilisant un réseau de neurones connu (`ResNet50`) qui a été préalablement entraîné sur un autre jeu de données (`imagenet`). Les variables correspondent aux valeurs des neurones de la dernière couche de ce réseau. Le réseau `ResNet50` est très performant pour reconnaître des objets sur une image et ces valeurs forment en quelque sorte une représentation en nombre réels que ce réseau déduit de chacune des images du jeu de données. Ces données se retrouvent dans le jeu de données *train-features.csv*.

Voici un résumé de ce que vous retrouverez dans les différents jeu de données :

```
str(data_id_train)
```

```
## Classes 'data.table' and 'data.frame':   1800 obs. of  2 variables:
##  $ ID            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ green_roof_ind: int  0 0 0 1 0 0 1 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(train_features, list.len = 8)
```

```
## Classes 'data.table' and 'data.frame':   1800 obs. of  2049 variables:
##  $ ID   : int  519 1594 1580 243 525 531 257 1231 1557 1543 ...
##  $ V1   : num  0 0.2064 0.0169 0 0.052 ...
##  $ V2   : num  1.0173 0.0743 0.0466 0.3934 0.3531 ...
##  $ V3   : num  0.0375 0 0.6051 0 0.109 ...
##  $ V4   : num  0.882 0.39 0.142 0.127 0.163 ...
##  $ V5   : num  0.1202 0.00332 0.01454 0.02275 0.03517 ...
##  $ V6   : num  0.00338 0.25047 0.03976 0 0.025 ...
##  $ V7   : num  4.3122 0.0589 0.9601 2.1706 0.9007 ...
##   [list output truncated]
##  - attr(*, ".internal.selfref")=<externalptr>
```

Livrables de la journée

Voici les moments importants de la journée:

- 15 h: Remise des projets (codes seulement)
- 15 h 30: Début des présentations
- 16 h 30: Remise des prix

Évaluation et prix

Les équipes ne seront pas évaluées en fonction des performances de leur modèle. En effet, chaque équipe devra plutôt faire une courte présentation (2 minutes **maximum**) de leur méthodologie. Les prix seront attribués de manière subjective par un groupe formé de 3 évaluateurs. Vous retrouverez ci-dessous les différents prix qui seront attribués par les juges. Les grilles d'évaluation ont été mises en annexe à ce document.

1. Le prix *Martin Luther King Jr.*: Un prix sera remis à l'équipe qui réalisé la meilleure **présentation orale** (voir les critères d'évaluation en annexe).
2. Le prix *Leonardo da Vinci*: Le groupe de juges devra attribuer un prix à l'équipe ayant présenté la solution la plus **créative**. Notez que la solution la plus créative ne doit pas nécessairement être la plus performante.
3. Le prix *Linus Torvalds*: Le groupe de juges devra également choisir l'équipe ayant remis le **code de meilleure qualité** (voir les critères en annexe).

Des prix de présence seront également remis au hasard à la fin de la journée.

MEETUP ML
Journée hackathon en assurance

Annexe

Grille d'évaluation: Prix *Martin Luther King Jr.*

Voici les critères d'évaluation pour la présentation qui devra être faite en équipe.

	Résultat
Critère 1: Présentation des résultats	
L'équipe a expliqué de manière claire et consise sa méthodologie.	/10
L'équipe a présenté de quelle manière ils ont validé les performances de leur modèle.	/5
Les résultats sont présentés clairement.	/5
Sous-total du critère 1:	**/20**
Critère 2: Capacité à bien présenter oralement	
L'équipe est en mesure de bien vulgariser sa démarche.	/10
La présentation est bien structurée et présentée dans un ordre logique.	/10
Tous les membres de l'équipe participent à la présentation.	/10
Les membres de l'équipe utilisent un vocabulaire et des termes faciles à comprendre.	/10
Sous-total du critère 2:	**/40**
Critère 3: Qualité du support visuel	
La méthodologie utilisée est présentée de manière visuelle et schématisée	/10
Le support visuel est simple, concis et apporte de la valeur aux explications.	/10
Sous-total du critère 3:	**/20**
Total:	**/80**

Critères d'évaluation: Prix *Leonardo da Vinci*

Il n'y pas de critères précis pour ce prix. Les juges auront à se consulter pour déterminer quelle équipe aura présenté la solution la plus créative.

Critères d'évaluation: Prix *Linus Torvalds*

Voici les critères d'évaluation qui seront utilisés pour la qualité du code:

- La structure générale du projet.
- L'indentation utilisée dans les programmes.
- Le choix des noms utilisés pour les fonctions, les variables, les méthodes et les classes.
- La constance dans le style de programmation.
- L'utilisation adéquate de commentaires.