

# Anticipez le retard de vol des avions

Base de données : Intégralité des vols commerciaux pour le transport de passagers aux Etats-Unis, année 2016

Source : [US bureau of transportation statistics](#)

Version figée : [lien de téléchargement](#)

# Objets de la mission

- Réaliser une analyse exploratoire
  - Relation entre les variables et leur impact sur les retards
  - Préparation de la base pour la modélisation
- Tester différentes approches
- Optimiser les hyperparamètres
- Choisir le modèle final
- Développer un site permettant de réaliser des prédictions sur les vols à venir

# Problématique

Quelles sont les informations dont le client a besoin pour organiser au mieux sa logistique ? Peut-on prédire les retards avec précision ? Est-ce que donner une simple prédiction sans un niveau de confiance élevé a un sens ?

- Quelles sont les variables qui impactent les retards ?
- Donner une estimation du retard attendu
- Fournir une probabilité d'occurrence par classe de retard

# Problématique

Mon approche :

- Donner une prédiction du retard pour tous les vols possibles aux Etats-Unis, en veillant à ce que l'approche adoptée pour l'année 2016 puisse être généralisable pour des vols dans le futur.
- Fournir des probabilités de prédiction pour plusieurs classes de retard afin de mesurer les risques de retard important et d'agir en conséquence.

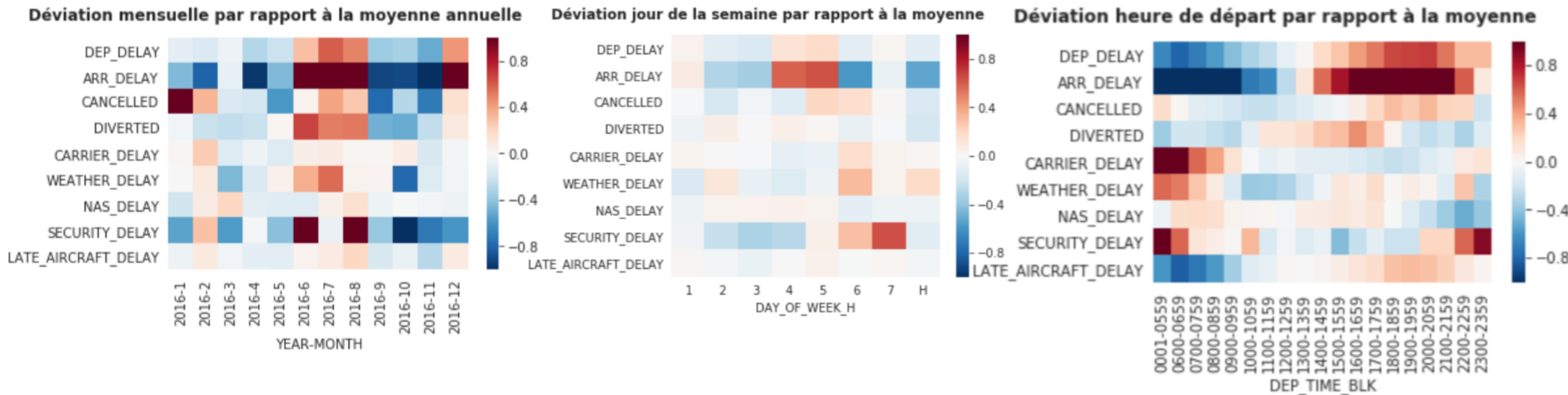
# 1. Analyse exploratoire

## a. Découverte des données

- Des lignes corrompues qui sont supprimées au moment de l'importation des données. Un total restant de plus de 5M de lignes.
- De nombreuses variables (64) :
  - Endogènes (35) décrivant le vol tel que planifié et disponible à l'utilisateur au moment de la réservation
  - Exogènes (27) décrivant les conditions réelles du vol
  - Une variable entre les 2 (TAIL\_NUM), et une variable de somme.

# 1. Analyse exploratoire

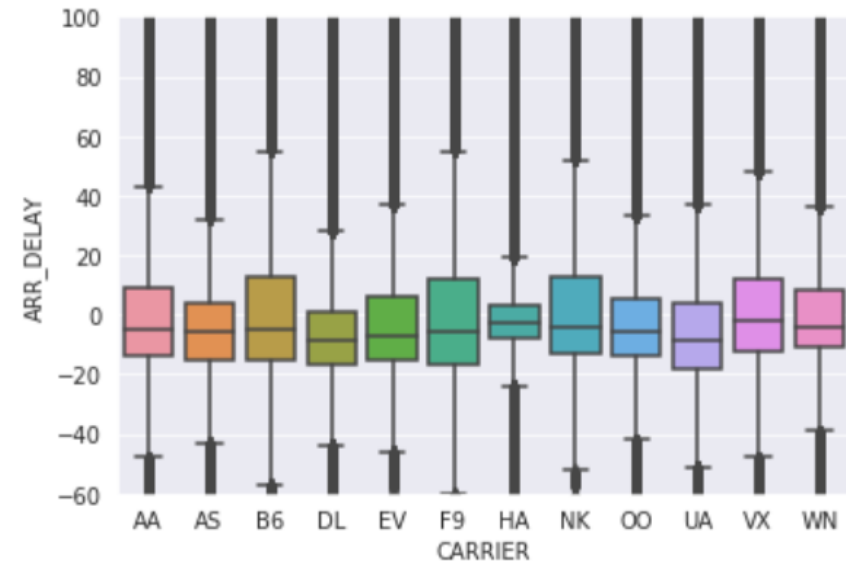
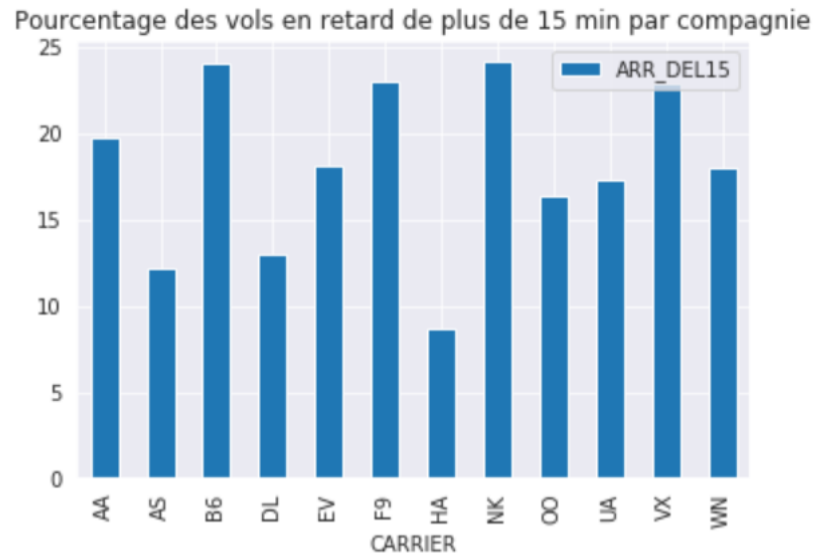
## b. Analyse des catégories temporelles



⇒ Des différences visibles entre les mois de l'année et aussi entre les heures de départ. Moins marquées entre les jours de la semaine.

# 1. Analyse exploratoire

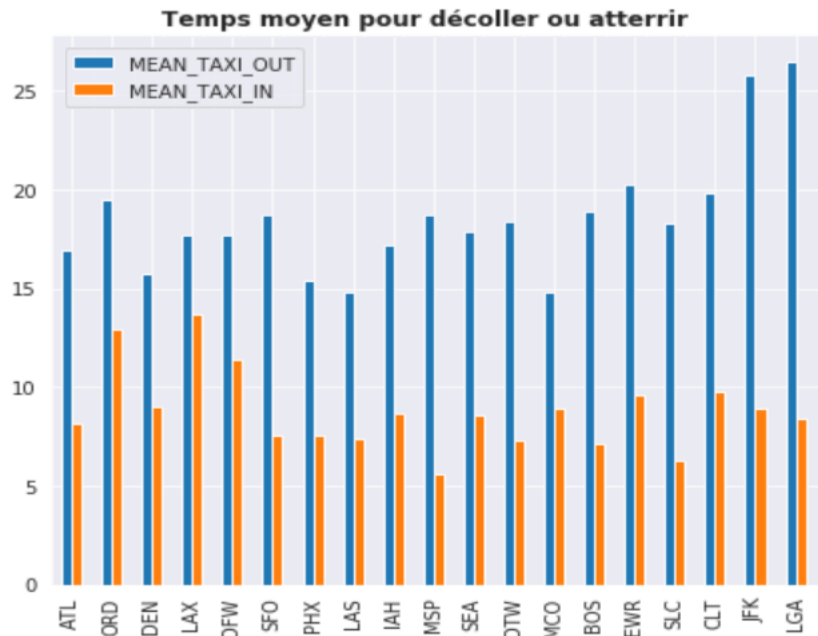
## c. Analyse du trajet : la compagnie aérienne



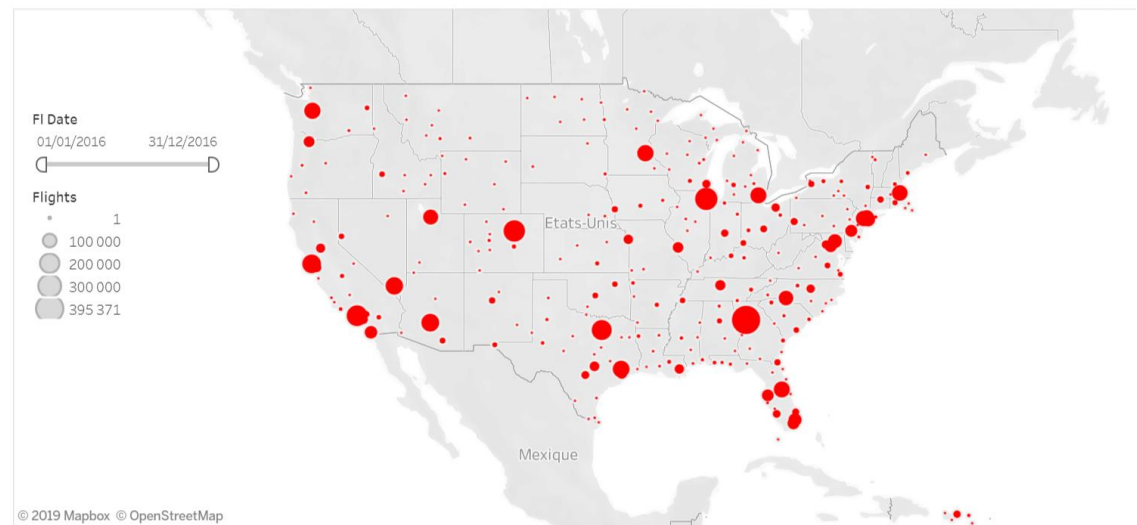
⇒ Les compagnies aériennes n'ont pas la même performance en terme de retards

# 1. Analyse exploratoire

## d. Analyse du trajet : aéroport de départ et d'arrivée



Répartition des destinations

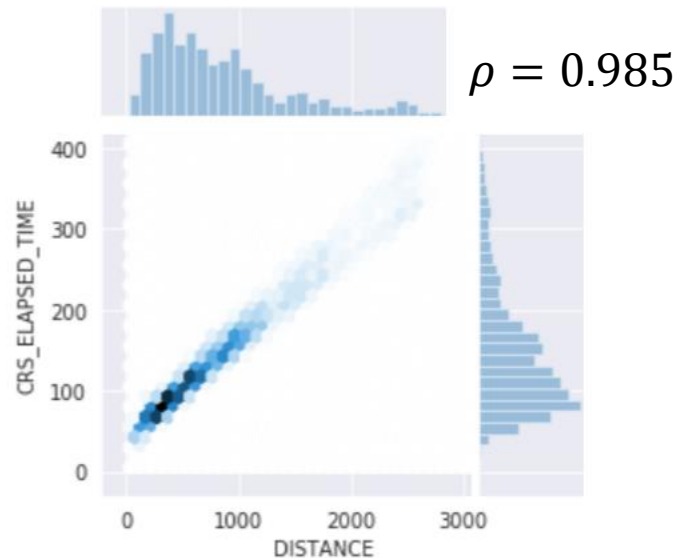


⇒ Des performances divergentes. Une vingtaine d'aéroports représente la majorité du trafic aérien



# 1. Analyse exploratoire

## e. Analyse du trajet : distance et temps de parcours



- Forte corrélation linéaire entre la distance et le temps de vol annoncé
- Des temps de vol annoncés pas toujours identiques pour un même trajet

Le retard est faiblement corrélé à la différence entre le temps annoncé pour un parcours et le temps habituel.

## 2. Modélisation

### a. Performance de base

- Suppression des outliers pour un aéroport par jour et pour une compagnie aérienne par jour : trop de retards ou d'annulation sur une journée
- Suppression des vols annulés, détournés ou avec plus de 3h de retard
- Suivi de 3 métriques pour le retard à l'arrivée :
  - Erreur quadratique moyenne (RMSE)
  - $R^2$
  - Erreur absolue moyenne

Baseline

RMSE: train=26.612, test=26.657

R2: train=0.000, test=-0.007

MAE: train=16.918, test=16.947

## 2. Modélisation

### b. Transformation des variables

- Variables catégorielles
  - *Trajet* : compagnie aérienne, aéroports de départ et de destination
  - *Date et heure* : mois, jour de la semaine (ou jour férié), heure de départ

⇒ One Hot Encoder

- Distance
- Différence entre le temps annoncé et le temps médian pour le trajet

⇒ Centrage-réduction

- Création de classes de retard (déséquilibrées) :

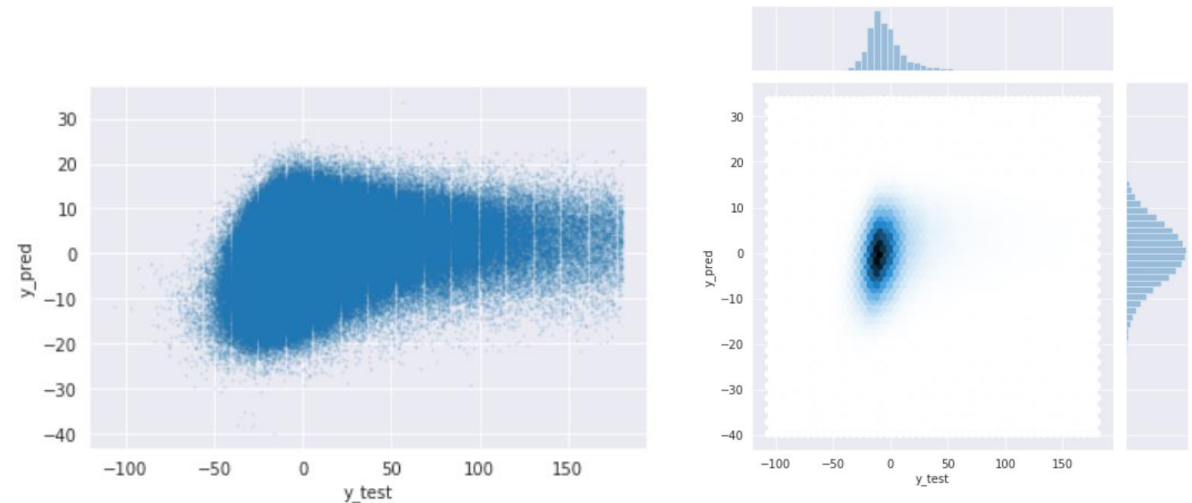
```
[-inf, 0]: 67.4%  
[1, 20]: 20.2%  
[21, 60]: 8.4%  
[61, 120]: 3.1%  
[121, 180]: 0.9%
```

## 2. Modélisation

### c. Régression linéaire

- Prédiction sans risque :

```
RMSE: train=25.804, test=26.029  
R2: train=0.060, test=0.039  
MAE: train=16.399, test=16.513
```



- La régularisation par Elastic Net n'a pas permis d'améliorer les résultats :

```
RMSE: train=26.404, test=26.470  
R2: train=0.015, test=0.015  
MAE: train=16.791, test=16.809
```

## 2. Modélisation

### d. Tentative de réduction des dimensions

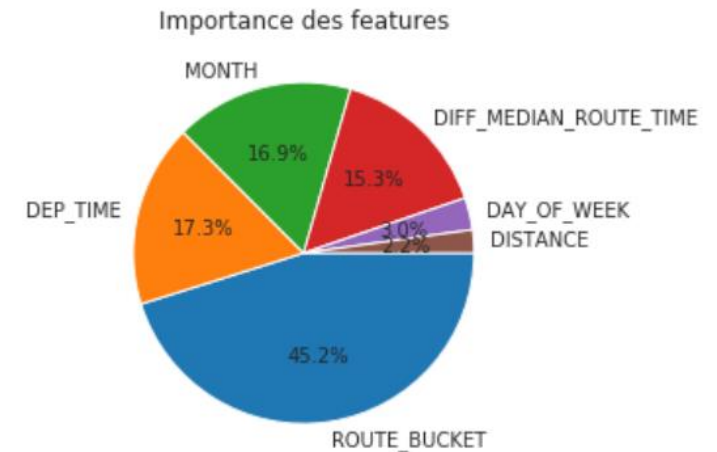
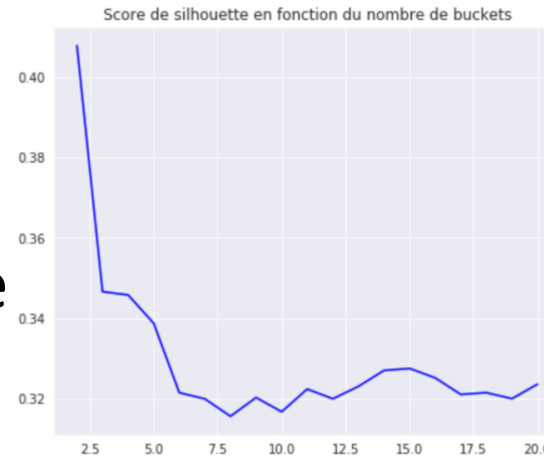
- Réduction par buckets :  
On regroupe compagnie, origine et destination d'après la moyenne des retards et l'écart-type

RMSE: train=25.803, test=25.890

R2: train=0.059, test=0.058

MAE: train=16.421, test=16.442

- Autres approches tentées mais moins performantes :
  - Réduction des variables catégorielles par leur retard moyen
  - Réduction par les quantiles des retards



## 2. Modélisation

### e. Test d'algorithmes et optimisation des paramètres

- Decision tree regressor
- XGBoost regressor
- XGBoost random forest regressor
- SGD Regressor

Optimisation bayésienne des paramètres sur un échantillon + cross-validation 5 folds sur le SGDRegressor et XGBRFRegressor. Ce dernier a la meilleure amélioration les données de test :

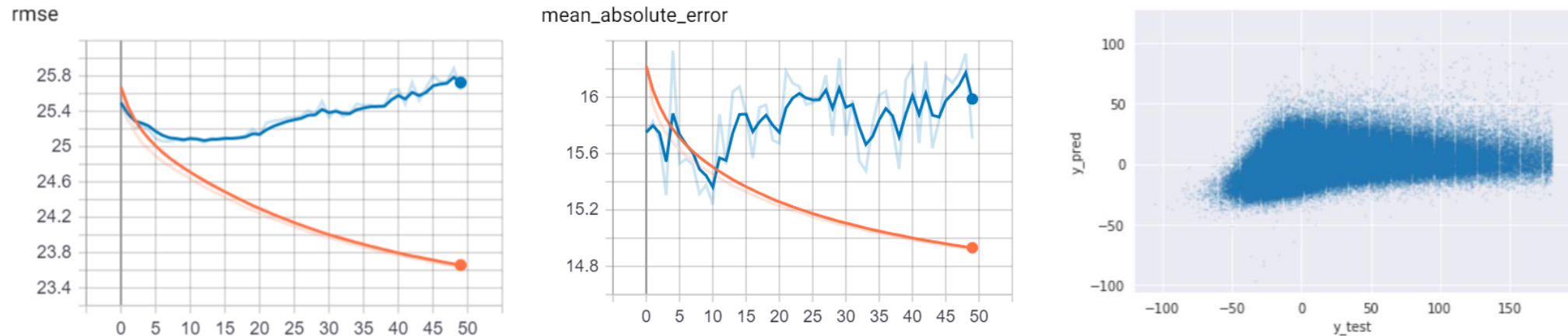
```
RMSE: train=25.616, test=25.702  
R2: train=0.073, test=0.071  
MAE: train=16.211, test=16.235
```

## 2. Modélisation

### f. Deep learning pour la régression

Succession de couches denses avec un nombre décroissant de neurones. Activation ReLU pour les couches intermédiaires.

Activation linéaire pour la dernière couche



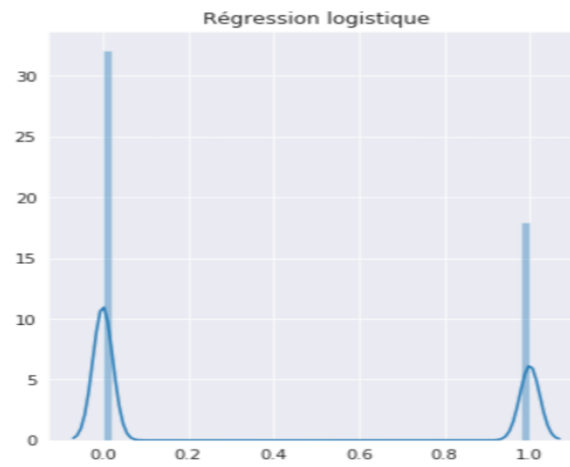
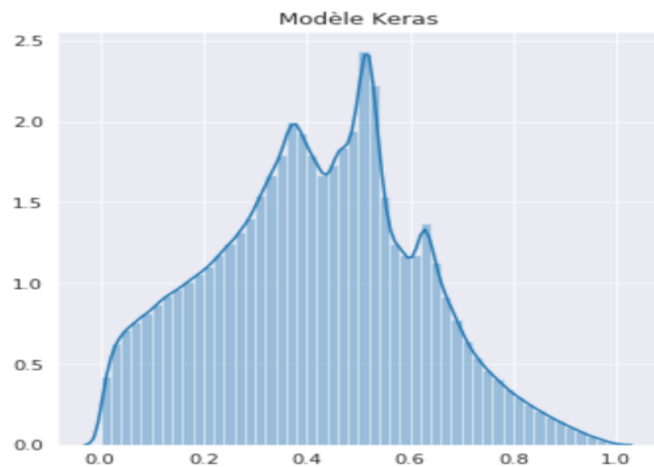
Amélioration des résultats :

RMSE: train=24.610, test=25.066  
R2: train=0.144, test=0.117  
MAE: train=15.139, test=15.376

## 2. Modélisation

### g. Deep learning pour la détection d'outliers

- Activation sigmoid pour la dernière couche
- Couches Dropout intermédiaires pour prévenir l'overfitting
- Rééquilibrage des classes : 50-50
- Accuracy à 67% vs 63% pour une régression logistique





## 2. Modélisation

### h. Deep learning pour la classification

- Activation softmax pour la dernière couche afin d'avoir des probabilités
- F1-score moyenne pondérée :
  - Deep learning : 0.58
  - 1 couche : 0.55
  - Prédiction constante : 0.55
  - Prédiction aléatoire : 0.51
- Comparaison entre les prédictions de classes et les probabilités associées :  $R^2 = 0.999$  vs 0.867 pour une couche

```
Class 0, [-inf, 0]:  
class distribution: 67.7%  
mean prediction: 68.0%  
mean prediction when 'DELAY_CLASS'==0: 71.7%  
mean prediction when 'DELAY_CLASS'!=0: 60.2%
```

```
Class 1, [1, 20]:  
class distribution: 20.1%  
mean prediction: 20.6%  
mean prediction when 'DELAY_CLASS'==1: 24.8%  
mean prediction when 'DELAY_CLASS'!=1: 19.6%
```

```
Class 2, [21, 60]:  
class distribution: 8.3%  
mean prediction: 7.9%  
mean prediction when 'DELAY_CLASS'==2: 12.1%  
mean prediction when 'DELAY_CLASS'!=2: 7.5%
```

```
Class 3, [61, 120]:  
class distribution: 3.0%  
mean prediction: 2.7%  
mean prediction when 'DELAY_CLASS'==3: 5.3%  
mean prediction when 'DELAY_CLASS'!=3: 2.6%
```

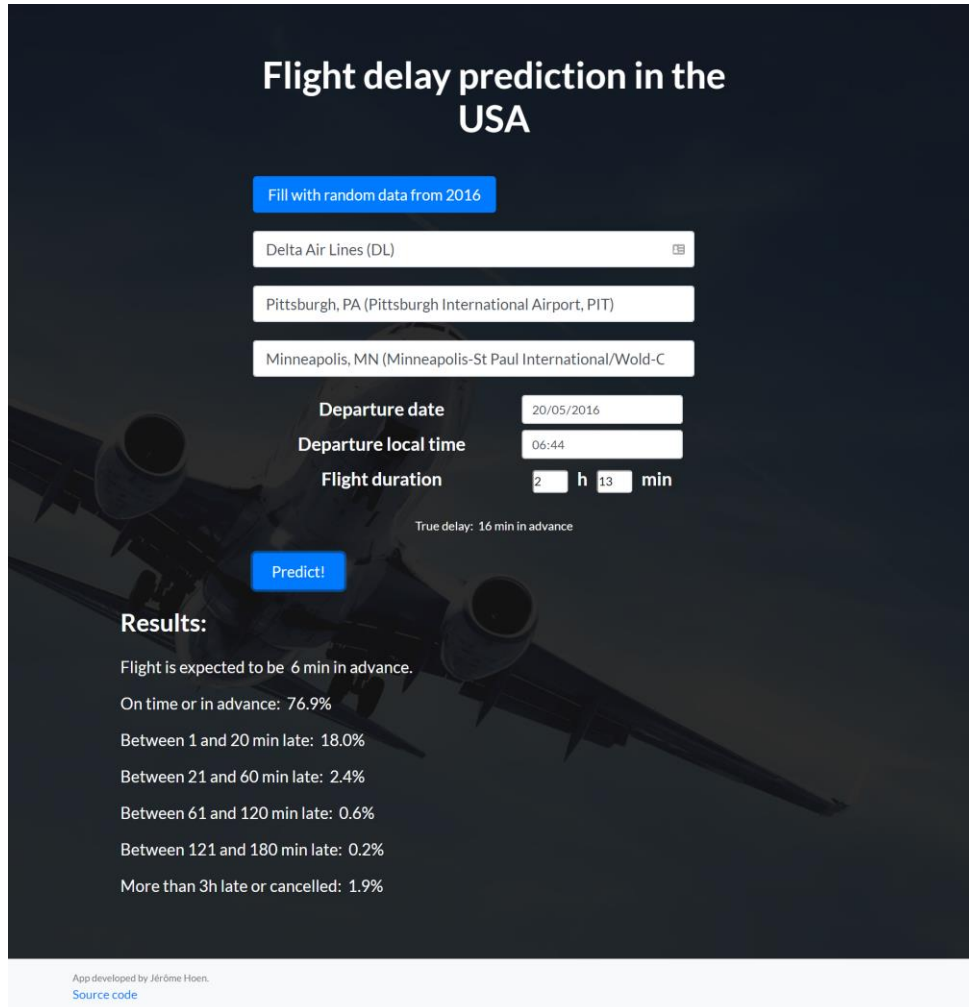
```
Class 4, [121, 180]:  
class distribution: 0.9%  
mean prediction: 0.8%  
mean prediction when 'DELAY_CLASS'==4: 1.8%  
mean prediction when 'DELAY_CLASS'!=4: 0.8%
```

# 3. Déploiement de la solution

## Préparation des fichiers

- Téléchargement des modèles, de l'encodeur et du scaler
- Création d'une matrice de distances entre tous les aéroports :
  - Projection conique [EPSG 2163](#) : (longitude, latitude) => (x, y)
  - Calcul des distances pour les liaisons non présentes dans la base
- Calcul du temps de trajet médian pour les liaisons manquantes d'après la distance

### 3. Déploiement de la solution



The screenshot shows a web application titled "Flight delay prediction in the USA". It features a dark background with a faint image of an airplane. The interface includes several input fields and buttons. At the top, there is a button labeled "Fill with random data from 2016". Below it are three text input fields: "Delta Air Lines (DL)", "Pittsburgh, PA (Pittsburgh International Airport, PIT)", and "Minneapolis, MN (Minneapolis-St Paul International/Wold-C)". Further down are two more input fields: "Departure date" with the value "20/05/2016" and "Departure local time" with the value "06:44". Below these is a "Flight duration" field showing "2 h 13 min". A small text label "True delay: 16 min in advance" is visible. A blue "Predict!" button is located below the input fields. Under the "Results:" section, the text "Flight is expected to be 6 min in advance." is displayed. Below this, a list of probabilities is shown: "On time or in advance: 76.9%", "Between 1 and 20 min late: 18.0%", "Between 21 and 60 min late: 2.4%", "Between 61 and 120 min late: 0.6%", "Between 121 and 180 min late: 0.2%", and "More than 3h late or cancelled: 1.9%". At the bottom left, small text reads "App developed by Jérôme Hoen." and "Source code".

Accessible à l'adresse :

<http://flightdelayprediction.herokuapp.com/>

- Permet de tester avec des données réelles de 2016
- Possibilité de créer des vols inédits
- Retard réel et retard calculé
- Probabilités des classes de retard rééquilibrées avec la probabilité d'avoir un outlier

# Conclusion

## a. Des résultats relativement décevants

- Au final, la régression donne des résultats très moyens même avec un modèle deep learning ( $R^2$  de 0.12)
- Explication :
  - Les compagnies ajustent le temps de vol en fonction de leurs propres prédictions pour éviter les retards en chaine
  - Les retards importants dus à des problèmes au caractère aléatoire
- Une amélioration toute de même sensible par rapport aux modèles les plus simples ( $R^2$  de 0.06 ou 0.07)

# Conclusion

## b. Régression contrebalancée par la classification

- Evaluation des risques : risque élevé d'un retard  $> 1h$  ?
  - Correspondance manquée
  - Risque de manquer un évènement à l'arrivée (livraison par exemple)
- En fonction des besoins et des conséquences d'un retard, action possible : comparaison avec d'autres vols similaires (autre date, heure ou compagnie)