

# SÉLECTION DES 8 VARIABLES (FEATURES) POUR LA PRÉDICTION DE RETARD

## 1. departure\_hour (Numérique - Heure de départ)

Justification: Les retards varient significativement selon l'heure de la journée.

L'analyse exploratoire a montré que certaines heures (matin tôt, soir) ont plus de retards.

## 2. is\_peak\_hours (Binaire - Heures de pointe)

Justification: Les heures de pointe (7-9h et 17-19h) augmentent le risque de retard en cascade dû au traffic aérien dense. Variable dérivée de departure\_hour.

## 3. flight\_duration\_min (Numérique - Durée du vol en minutes)

Justification: Les vols longs ont potentiellement plus de variables pouvant causer des retards (météo sur longer trajet, consommation carburant, fatigue équipage).

## 4. load\_factor\_pct (Numérique - Taux de remplissage)

Justification: Un avion très rempli peut avoir des temps d'embarquement/débarquement

plus longs, notamment pour les classes avec plus de bagages (Affaires, Première).

## 5. passengers (Numérique - Nombre de passagers)

Justification: Plus de passagers = plus de chances de retards individuels (retard au départ, bagages, demandes spéciales).

## 6. distance\_km (Numérique - Distance en km)

Justification: Les vols longs traversent potentiellement plus de zones météorologiques différentes et ont plus de contraintes opérationnelles.

- Source: Colonnes existantes

## 7. is\_weekend (Binaire - Week-end)

Justification: Le trafic et les comportements sont différents le week-end.  
Plus de voyageurs loisirs, possiblement différents patterns de retards.

## 8. is\_international (Binaire - Vol international)

Justification: Les vols internationaux ont des procédures additionnelles (contrôles douaniers, immigration, correspondance internationale) qui peuvent causer des retards.

# Encodage des variables catégorielles

## DIFFÉRENCE ENTRE LABEL ENCODING ET ONE-HOT ENCODING

### LABEL ENCODING:

- Chaque catégorie est remplacée par un nombre entier (0, 1, 2, ...)
- Avantages:

\* Ne crée pas de nouvelles colonnes, efficace en mémoire

\* Conserve l'ordre pour les variables ordinaires

- Inconvénients:

\* Implique un ordre qui peut être mal interprété par certains modèles

\* Les modèles linéaires peuvent considérer cet ordre comme significatif

- Usage recommandé: Variables ordinaires (ex: Petit=0, Moyen=1, Grand=2)

#### ONE-HOT ENCODING:

- Chaque catégorie devient une nouvelle colonne binaire (0 ou 1)
- Avantages:
  - \* Pas d'implication d'ordre, représentation claire pour les modèles
  - \* Chaque catégorie est traitée indépendamment
  - Inconvénients:
    - \* Crée beaucoup de colonnes si nombreuses catégories (curse of dimensionality)
    - \* Peut créer de la redondance pour les variables à 2 modalités
    - Usage recommandé: Variables nominales sans ordre intrinsèque

## RAISON D'UTILISER DEUX ENSEMBLES SÉPARÉS POURQUOI DIVISER LES DONNÉES?

### 1. Estimer la performance RÉELLE du modèle:

- Si on teste sur les mêmes données qu'on a utilisées pour entraîner, le modèle aura l'air plus performant qu'il ne l'est en réalité
- On dit souvent qu'un modèle "triche" s'il voit les réponses pendant l'examen

### 2. Éviter le sur-apprentissage (overfitting):

- Le modèle peut apprendre des patterns spécifiques aux données d'entraînement
- Ces patterns ne se généraliseront pas aux nouvelles données
- La séparation permet de détecter ce problème

### 3. Valider que le modèle généralise bien:

- Les données de test simulent des données futures non vues

- Si les performances sont bonnes sur le test, le modèle est robuste

#### 4. Évaluer différents modèles de manière équitable:

- Permet de comparer plusieurs modèles sur les mêmes données

## QUEST-CE QUE LE DATA LEAKAGE?

Definition: Quand des informations du "futur" "fuient" dans les données d'entraînement,

le modèle apprend des patterns qu'il ne devrait pas voir en pratique.

### EXEMPLES DE DATA LEAKAGE:

- Utiliser delay\_minutes pour prédire is\_delayed (c'est exactement la même variable!)
- Normaliser les données AVANT de diviser (calculer moyenne/écart-type sur tout le dataset)
- Inclure des variables qui contiennent intrinsèquement l'information à prédire
- Utiliser des données futures (ex: prévoir des ventes avec les ventes du mois prochain)

### COMMENT L'ÉVITER ICI?

#### 1. On utilise UNIQUEMENT les features qui ne contiennent pas d'information directe sur le retard

- is\_delayed est créé à partir de delay\_minutes, mais on n'utilise pas delay\_minutes comme feature

#### 2. On split les données AVANT toute transformation

- Les encodeurs sont conçus pour ne pas voir les données de test

#### 3. On fit les encodeurs/scalers UNIQUEMENT sur le training set

- Cela garantit que le test set reste "invisible" pendant l'entraînement