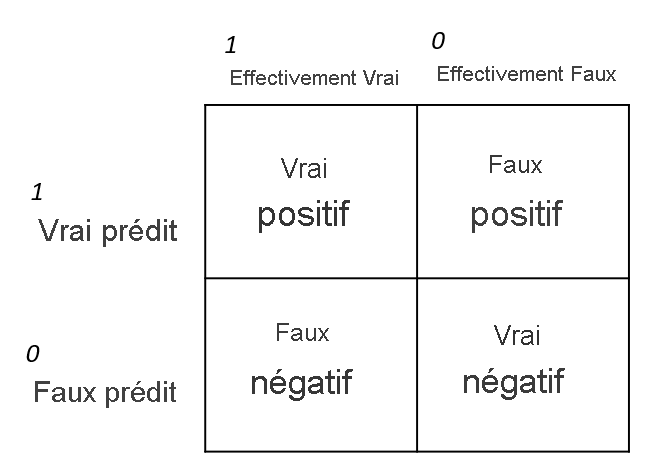
**Matrice de confusion**

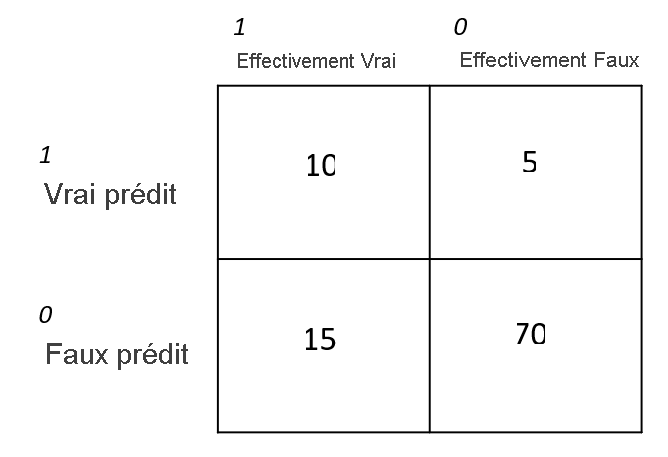
La matrice de confusion est un outil utilisé pour évaluer la qualité des prédictions d’un modèle de classification. Elle compare les étiquettes prédites aux étiquettes réelles.

Dans le cas d’un modèle de classification binaire où vous prédisez une valeur parmi deux valeurs possibles, la matrice de confusion est une grille de 2x2 qui montre le nombre de valeurs prédites et de valeurs réelles pour les classes **1** et **0**. Elle catégorise les résultats du modèle en quatre types de résultats. Avec notre exemple sur le diabète, ces résultats peuvent se présenter comme ceci :

* *Vrai positif* : le modèle prédit que le patient a le diabète et le patient a effectivement le diabète.
* *Faux positif* : le modèle prédit que le patient a le diabète, mais le patient n’a en réalité pas le diabète.
* *Faux négatif* : le modèle prédit que le patient n’a pas le diabète, mais le patient a en réalité le diabète.
* *Vrai négatif* : le modèle prédit que le patient n’a pas le diabète, et le patient n’a effectivement pas le diabète.



Supposons que vous avec des données pour 100 patients. Vous créez un modèle qui prédit qu’un patient n’a pas le diabète dans 15 % des cas : il *prédit* donc que 15 personnes ont le diabète et *prédit* que 85 personnes n’ont pas de diabète. En réalité, supposons que 25 personnes ont *effectivement* le diabète et que 75 personnes n’ont *effectivement* pas le diabète. Ces informations peuvent être présentées dans une matrice de confusion comme celle ci-dessous :



Pour un modèle de classification multiclasse (avec plus de deux classes possibles), la même approche est utilisée pour représenter sous forme de tableau chaque combinaison possible de nombres de valeurs réelles et prédites. Ainsi, un modèle avec trois classes possibles produit une matrice 3x3 avec une ligne diagonale de cellules où les étiquettes prédites et réelles correspondent.

Les métriques qui peuvent être dérivées de la matrice de confusion sont les suivantes :

* **Justesse** : nombre de prédictions correctes (vrais positifs + vrais négatifs) divisé au par le nombre total de prédictions.
* **Précision** : nombre de cas classifiés comme positifs qui sont réellement positifs : le nombre de vrais positifs divisé par (le nombre de vrais positifs plus les faux positifs).
* **Rappel** : fraction des cas positifs correctement identifiés : le nombre de vrais positifs divisé par (le nombre de vrais positifs plus les faux négatifs).
* **Score F1** : métrique globale combinant essentiellement la précision et le rappel.

Parmi ces métriques, la *justesse* peut être la plus intuitive. Cependant, vous devez être prudent dans l’utilisation de la justesse comme mesure de la performance d’un modèle. En utilisant le modèle qui prédit que 15 % des patients sont diabétiques, alors qu’en fait 25 % des patients sont diabétiques, nous pouvons calculer les métriques suivantes :

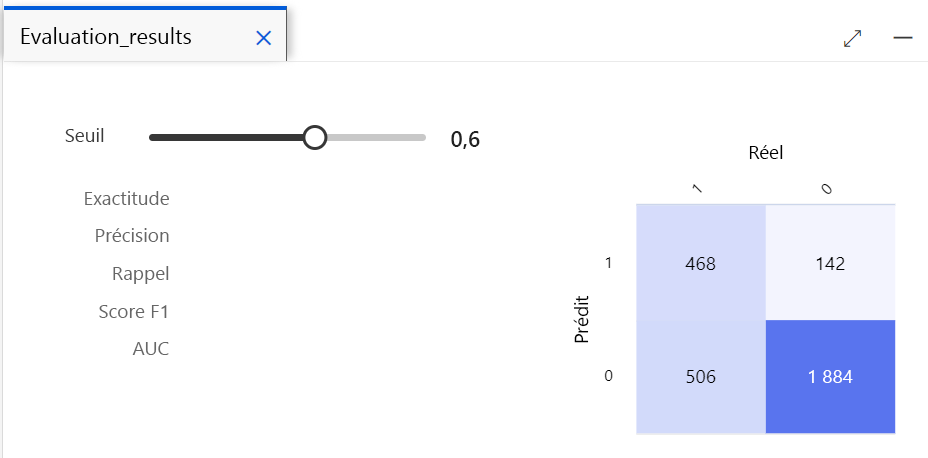
La *justesse* du modèle est la suivante : (10+70) / 100 = 80 %.

La *précision* du modèle est : 10 / (10+5) = 67 %.

Le *rappel* du modèle est : 10 / (10+15) = 40 %.

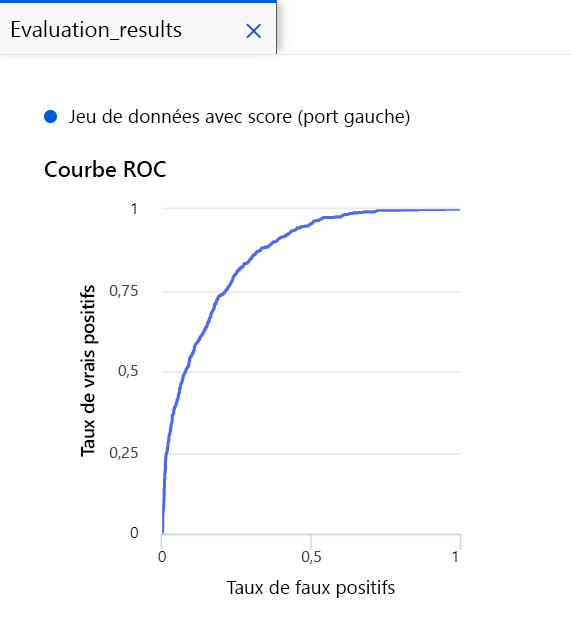
**Choix d’un seuil**

Un modèle de classification prédit la probabilité pour chaque classe possible. En d’autres termes, le modèle calcule une probabilité pour chaque étiquette prédite. Dans le cas de ce modèle de classification binaire, la probabilité prédite est une valeur comprise entre 0 et 1. Par défaut, une probabilité prédite *supérieure ou égale* à 0,5 donne une prédiction de classe 1, tandis qu’une prédiction *inférieure* à ce seuil signifie que la probabilité d’une prédiction *négative* est supérieure (n’oubliez pas que la somme des probabilités pour toutes les classes est égale à 1). La classe prédite sera donc 0.

Le concepteur dispose d’un *curseur de seuil* utile pour examiner la façon dont les performances du modèle changent en fonction du seuil défini. 

**Courbe ROC et métrique AUC**

Le *rappel*, également désigné **taux de vrais positifs**, a une métrique correspondante nommée **Taux de faux positifs**, qui mesure le nombre de cas négatifs incorrectement identifiés comme positifs par rapport au nombre de cas négatifs réels. Tracer ces métriques les unes par rapport aux autres pour chaque valeur de seuil possible comprise entre 0 et 1 produit une courbe, appelée **courbe ROC** (ROC signifie *caractéristique de fonctionnement du récepteur*, mais la plupart des scientifiques des données l’appellent simplement courbe ROC). Dans un modèle idéal, la courbe s’étendrait jusqu’en haut du côté gauche (et poursuivrait sa progression au-delà), couvrant toute la zone du graphique. Plus la *zone sous la courbe* de la métrique **AUC** (toute valeur comprise entre 0 et 1) est étendue, plus le modèle est performant. Vous pouvez examiner la courbe ROC dans **Résultats de l’évaluation**.



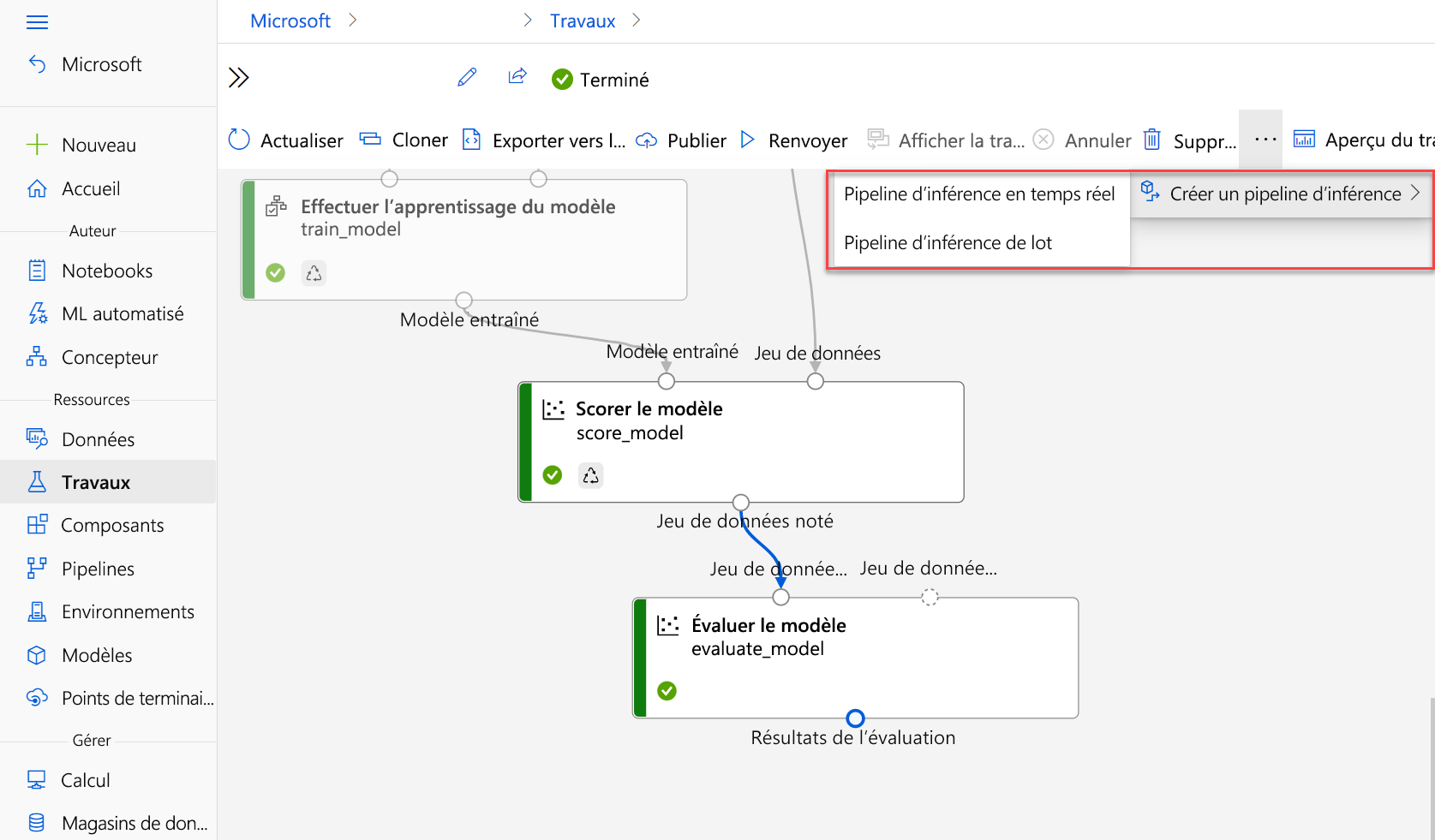
**Déployer un service prédictif**

Vous avez la possibilité de déployer un service utilisable en temps réel. Pour automatiser votre modèle dans un service qui effectue des prédictions continues, vous devez créer et déployer un pipeline d’inférence.

**Pipeline d’inférence**

Pour déployer votre pipeline, vous devez d’abord convertir le pipeline d’entraînement en pipeline d’inférence en temps réel. Ce processus supprime les composants de formation et ajoute les entrées et sorties de service web pour gérer les demandes.

Le pipeline d’inférence effectue les mêmes transformations de données que le premier pipeline pour les *nouvelles* données. Ensuite, il utilise le modèle entraîné pour *inférer*, ou prédire, des valeurs d’étiquette en fonction de leurs caractéristiques. Ce modèle va constituer la base d’un service prédictif que vous pouvez publier en vue de son utilisation par des applications.

Vous pouvez créer un pipeline d’inférence en sélectionnant le menu au-dessus d’un travail terminé. 

# Contrôle des connaissances

Effectué200 XP

* 3 minutes

Top of Form

1.

Vous utilisez le concepteur Azure Machine Learning afin de créer un pipeline d’entraînement pour un modèle de classification binaire. Vous avez ajouté un jeu de données contenant des caractéristiques et des étiquettes, un module Forêt d’arbres décisionnels à deux classes et un module Entraîner le modèle. Vous prévoyez d’utiliser les modules Noter le modèle et Évaluer le modèle pour tester le modèle entraîné avec un sous-ensemble du jeu de données qui n’a pas été utilisé pour l’entraînement. Quel autre module devez-vous ajouter ?



Joindre des données



Fractionner les données

Correct. Utilisez un module Split Data pour diviser un jeu de données en sous-ensembles d’entraînement et de validation de façon aléatoire.



Sélectionner des colonnes dans le jeu de données

Réponse incorrecte. Utilisez un module Split Data pour découper un jeu de données en sous-ensembles de test et de validation de façon aléatoire.

2.

Vous utilisez un pipeline du concepteur Azure Machine Learning pour entraîner et tester un modèle de classification binaire. Vous passez en revue les métriques de performance du modèle dans un module Evaluate Model et remarquez que son score AUC est 0.3. Que pouvez-vous en conclure à propos du modèle ?



Le modèle peut expliquer 30 % de la variance entre les étiquettes réelles et prédites.



Le modèle fournit une prédiction juste pour 70 % des cas de test.



Le modèle est moins performant qu’une estimation aléatoire.

Correct. Vous pouvez vous attendre à un AUC de 0.5 avec une prédiction aléatoire d’un modèle binaire.

3.

Vous utilisez le concepteur Azure Machine Learning afin de créer un pipeline d’entraînement pour un modèle de classification. Que devez-vous faire avant de déployer le modèle en tant que service ?



Créer un pipeline d’inférence à partir du pipeline d’entraînement

Correct. Vous devez créer un pipeline d’inférence pour effectuer un déploiement en tant que service.



Ajouter un module Evaluate Model au pipeline d’entraînement



Cloner le pipeline d’entraînement avec un nom différent

Bottom of Form

# Comprendre les étapes du clustering

## Évaluer les performances

Après l’apprentissage d’un modèle, il est important d’évaluer ses performances. Il existe de nombreuses métriques et méthodologies de performances pour évaluer la qualité des prédictions d’un modèle. Vous pouvez examiner les mesures d’évaluation sur la page du travail terminé en cliquant avec le bouton droit sur le composant **Évaluer le modèle** .

Une fois l’exécution de l’expérience terminée, sélectionnez **Détails du travail**. Cliquez avec le bouton droit sur le module **Évaluer le modèle** et sélectionnez **Aperçu des données**, puis **Résultats de l’évaluation**. Ces métriques peuvent aider les scientifiques des données à évaluer la façon dont le modèle sépare les clusters. Elles incluent une ligne de métriques pour chaque cluster ainsi qu’une ligne de synthèse pour une évaluation combinée. Les métriques de chaque ligne sont les suivantes :

* **Average Distance to Other Center** : indique la distance, en moyenne, qui sépare chaque point dans le cluster des centroïdes de tous les autres clusters.
* **Average Distance to Cluster Center** : indique la distance, en moyenne, qui sépare chaque point dans le cluster du centroïde de ce cluster.
* **Number of Points** : nombre de points affectés au cluster.
* **Maximal Distance to Cluster Center** : Distances maximales entre chaque point et le centroïde du cluster de ce point. Si ce nombre est élevé, cela peut indiquer une grande dispersion dans le cluster. Cette statistique, combinée à la métrique **Average Distance to Cluster Center**, vous aide à déterminer la *répartition* du cluster.