ENSIAS

جامعة محمد الخامس بالرباط
Université Mohammed V de Rabat

École Nationale Supérieure d’Informatique et d’Analyse des Systèmes

May 2024

Data Warehouse Assignment

ForestAI - Building a Data Warehouse for Enhanced Model Validation and Client Insights

Elaboated by:
Amin BENALI, BI&A
Oumaima GHAZOUAN, BI&A
Yahya OUTGUOUGUA, BI&A

Jury:
Mme. BENHIBA LAILA

- Year school : 2023/2024 -

SUMMARY

This project developed a data warehouse and BI solution to address ForestAI's needs for streamlined model validation, improved data analysis, and interactive client dashboards. A data model was designed to efficiently store relevant metrics, and the solution was implemented using SSIS for data loading, an SSAS cube for analysis, and BI tools for visualization. This improved system empowers ForestAI with faster validation, deeper insights, and effective client presentations.

ABSTRACT

Ce projet a permis de développer un data warehouse et une solution BI complète pour répondre aux besoins de ForestAI en matière de rationalisation de la validation des modèles, d'amélioration de l'analyse des données et de tableaux de bord interactifs pour les clients. Un modèle de données a été conçu pour stocker efficacement les indicateurs pertinents. La solution a été implémentée à l'aide de SSIS pour le chargement des données, d'un cube SSAS pour l'analyse et d'outils BI pour la visualisation. Ce système amélioré permet à ForestAI de réaliser des validations plus rapides, d'obtenir des analyses plus approfondies et de présenter ses résultats aux clients de manière convaincante.

CONTENTS

LIST OF FIGURES

GENERAL INTRODUCTION

This report presents the design and implementation of a data warehouse solution for ForestAI, a company specializing in applying Data Science to predict forest inventories. The primary objective of this project is to overcome the current limitations of ForestAI's platform by improving the benchmarking process of the models, which is important for validating model quality and building client trust in ForestAI's products.

ForestAI uses climate, geospatial, and customer process data To effectively pinpoint possible buying and harvestable areas in forests with fewer field visits. However, the current model validation process is fragmented and lacks standardization, leading to inefficiencies and poor user experience.

To address these issues, this project aims to create a standardized data pipeline that integrates data from various sources (ground truth, company predictions, client models, and competitor models) into a common destination. This will enable quick data analysis for technical and business insights while ensuring consistent data quality. Additionally, the introduction of intermediate storage is intended to improve overall system performance.

The project involves designing a data warehouse and creating interactive dashboards, thus enhancing visualization and improving the user experience for ForestAI's clients. This report details the conceptual choices, data modeling, implementation steps, and provides a demonstration of the developed solution.

CHAPTER 1

INTRODUCTION TO GENERALITIES

1.1 Introduction

Forest inventory assessment is essential for sustainable forest management. ForestAI exploits the advantages of data science to predict key forest metrics like tree volume and species composition. However, streamlining model validation, improving data analysis, and providing client-friendly insights remain challenges. This project outlines the development of a data warehouse to address these needs and empower ForestAI with a robust data analysis platform.

1.2 Project Background

ForestAI's current workflow involves data acquisition, model development, and model validation for forest inventory prediction. Data resides in various sources, making analysis overly complicated and obstructive in-depth validation. Additionally, presenting insights to clients often requires manual data manipulation, leading to inefficiencies.

1.3 Project Objectives

This project aims to:

- Establish a centralized data warehouse to integrate and store forest inventory data from different sources.
- Design interactive dashboards for efficient model validation and data visualization.
- Provide clients with user-friendly access to key metrics and forest inventory insights.

1.4 Expected Outcomes

A successful project will achieve the following outcomes:

Improved model validation efficiency: Easier access to historical data and streamlined analysis will facilitate faster and more robust model validation.

Enhanced data analytics capabilities: The data warehouse will enable advanced data exploration and analysis, leading to deeper insights into forest health and composition.

Actionable client insights: Interactive dashboards will empower clients to readily access and comprehend key forest inventory data, aiding decision-making.

Increased efficiency: Streamlined workflows and automated reporting will free up resources for further innovation and client support.

1.5 Project Scope

This project focuses on building a data warehouse that integrates internal and potentially external forest inventory data sources. The scope includes:

- Data modeling and warehouse design
- Data extraction, transformation, and loading (ETL) processes
- Interactive dashboard development for model validation and client communication
- Data security and user access management

1.6 Conclusion

Building a data warehouse is an investment in ForestAI's future. It will enhance model development, empower data analysis, and provide valuable insights to clients. This project will lay the foundation for a data-driven approach to forest inventory prediction, leading to better decision-making and improved forest management practices.

CHAPTER 2 DATA MODELING

2.1 Introduction

Data modeling is an important step in designing a data warehouse as it structures information in a way that optimizes analysis and reporting. In this chapter, we will detail the process of creating the data modeling schema by identifying the necessary dimensions and facts to meet the needs of the dashboards and reports of the BI solution.

2.2 Reverse Engineering and Requirements

The first step in designing our data warehouse involves reverse-engineering the existing reports and understanding the current metrics, dimensions, and visualizations used. This process helps to identify the gaps and additional requirements needed to create a more comprehensive and efficient data warehouse.

The existing reports primarily focus on various performance metrics such as[1][2]:

- Y-MEANS (Mean of observed values)

$$y_{\text{mean}} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Figure 2.1: Formule de Y-MEANS

- PRED-MEANS (Mean of predicted values)

$$\text{pred}_{\text{mean}} = \frac{1}{n} \sum_{i=1}^{n} \hat{y}_i$$

Figure 2.2: Formule de PRED-MEANS

- RMSE (Root Mean Square Error)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Figure 2.3: Formule de RMSE

- MSE (Mean Square Error)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Figure 2.4: Formule de MSE

- MAE (Mean Absolute Error)

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right|$$

Figure 2.5: Formule de MAE

- MNAE (Mean Normalized Absolute Error)

$$NMAE_1 = \frac{MAE}{mean(actual\ values)}$$

Figure 2.6: Formule de MNAE

- NRMSE (Normalized Root Mean Square Error)[4]

$$NRMSE = \frac{RMSE}{\bar{y}}$$

Figure 2.7: Formule de NRMSE

- Bias

$$\text{bias} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)$$

Figure 2.8: Formule de bias

- NBIAS (Normalized Bias)

$$\text{nbias} = \frac{\text{bias}}{y_{\text{mean}}}$$

Figure 2.9: Formule de Nbias

These metrics are essential for evaluating the performance of different models and understanding the accuracy of predictions.

We aim to create visualizations that highlight these performance metrics, allowing for easy comparison and analysis. The dashboards will display scatter plots comparing observed values and predicted values, efficient tables that shows the evolution of these measurement.

2.3 Metrics List

The second step is to identify the metrics that will be visualized in the dashboards and reports. Based on research on existing MLOps solutions, we have determined the following metrics:

- Total Volume (m3/ha)
- Total Timber (m3/ha)
- Average Height (m)
- Total Basal Area (m2/ha)
- Average DBH (cm)
- Pine Volume (m3/ha)
- Pine Timber (m3/ha)
- Average Pine Height (m)
- Pine Basal Area (m2/ha)
- Average Pine DBH (cm)
- Spruce Volume (m3/ha)
- Spruce Timber (m3/ha)
- Average Spruce Height (m)
- Spruce Basal Area (m2/ha)
- Average Spruce DBH (cm)
- Deciduous Volume (m3/ha)
- Deciduous Timber (m3/ha)
- Average Deciduous Height (m)
- Deciduous Basal Area (m2/ha)
- Average Deciduous DBH (cm)

2.4 Dimensions

Dimensional modeling is a design technique used to structure the data warehouse schema. It involves creating fact tables that store quantitative data and dimension tables that provide context to the facts. This approach supports efficient querying and reporting.

Dimensions Identified:

- Time Dimension: Captures date-related information to allow analysis over different time periods.
- Location Dimension: Provides geographical context, including country, region, and specific plot details.
- Metric Dimension: Defines the various forestry metrics being measured and their units.
- Tree Species Dimension: Categorizes data by tree species, such as Pine, Spruce, and Deciduous.
- Model Dimension: Details the predictive models used, including their versions and descriptions.
- Client Dimension: Contains client-specific information, such as client names and regions.

2.5 Granularity of Facts

In our project, the granularity of the fact tables has been carefully determined to ensure that the data captures the required level of detail for each type of fact: Observed Value, Client Estimate, and Model Prediction. We have chosen these specific fact tables because **all other metrics can be calculated from these core data points.** We have determined the following levels of granularity for each fact:

1. Observed Value

    a By time period (day, month, year)

    b By location (plot, region)

    c By tree species

    d By metric

2. Client Estimate

    a By time period

    b By location

    c By tree species

    d By metric

    e By client

3. Model Prediction

   a By time period

   b By location

   c By tree species

   d By metric

   e By model

Here is the table that summarizes these granularities.

Facts	**Time**	**Location**	**Metric**	**Tree Species**	**Model**	**Client**
Observed Value	x	x	x	x		
Client Estimate	x	x	x	x		x
Model Prediction	x	x	x	x	x	

2.6 Fact tables list

Fact tables store the measurable, quantitative data. Based on the previous section, each fact is analysable by a different set of dimensions, therefore we have three fact tables:

1. Observed Values Fact Table (Ground Truth):
   - Columns: Time ID, Location ID, Metric ID, Species ID, Observed Value
   - Granularity: Individual plot measurements over time for different metrics and species.
2. Client Estimates Fact Table:
   - Columns: Time ID, Location ID, Metric ID, Species ID, Client ID, Client Estimate
   - Granularity: Client-provided estimates for various metrics and species over time and location.
3. Model Predictions Fact Table:
   - Columns: Time ID, Location ID, Metric ID, Species ID, Model ID, Model Prediction, Model Performance
   - Granularity: Model predictions and performance indicators for different metrics, species, and locations over time.

2.7 Data Modeling Schema

Based on the identified dimensions and facts, we have designed a star schema to structure the data. This schema includes:

- A main fact table for each type of fact (Observed Value, Client Estimate, Model Prediction).
- Associated dimension tables for time, location, metrics, tree species, models, and clients.

Additionally, we choose the **star schema** as database schema for our data mart because **our dimensions do not involve hierarchies** . Each dimension is treated as an independent entity, directly connected to the fact tables. This lack of hierarchies simplifies the design and usage, ensuring that users can easily query and analyze the data without navigating complex hierarchies.

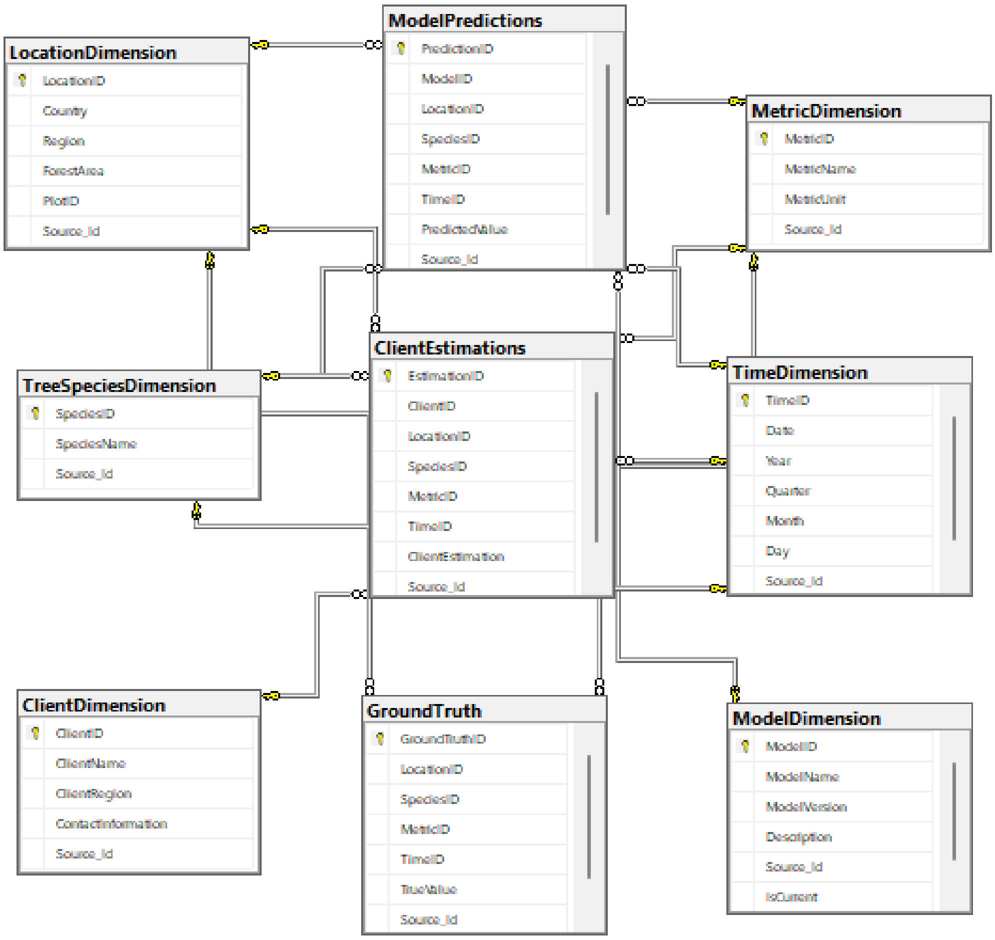


Figure 2.10: Forest AI datawarehouse shema

2.8 Conclusion

The design of the data warehouse for forestry metrics analysis provides a comprehensive framework to capture, store, and analyze various metrics, supporting robust reporting and performance monitoring. By utilizing dimensional modeling and a star schema, the data warehouse enables efficient querying and detailed analysis across multiple dimensions. The architecture ensures scalability and performance, supporting the business needs for accurate and timely insights into forestry metrics and model performance. This design will facilitate better decision-making and enhance the ability to track and improve predictive models.

CHAPTER 3

DATA WAREHOUSE IMPLEMENTATION

3.1 Introduction

In this section, we will delve into the details of implementing our data warehouse, a crucial step in our project. We will first explore the data sources used, describing their nature and structure. Next, we will discuss the ETL (Extract, Transform, Load) process that was implemented to extract data from sources, prepare it, and load it into our data warehouse environment. Finally, we will focus on the design and construction of the cube using SSAS (SQL Server Analysis Services), highlighting design choices, dimensions, facts, and aggregated measures available for analysis. This section provides a comprehensive overview of how our data warehouse was practically implemented, from the flow of raw data to the creation of a robust analytical environment.

3.2 Data Source

We have established a critical data source in the form of a relational database. This database was created and populated using Microsoft SQL Server Management Studio, a robust and efficient tool for database management.

Figure 3.1: Microsoft SQL Server Logo

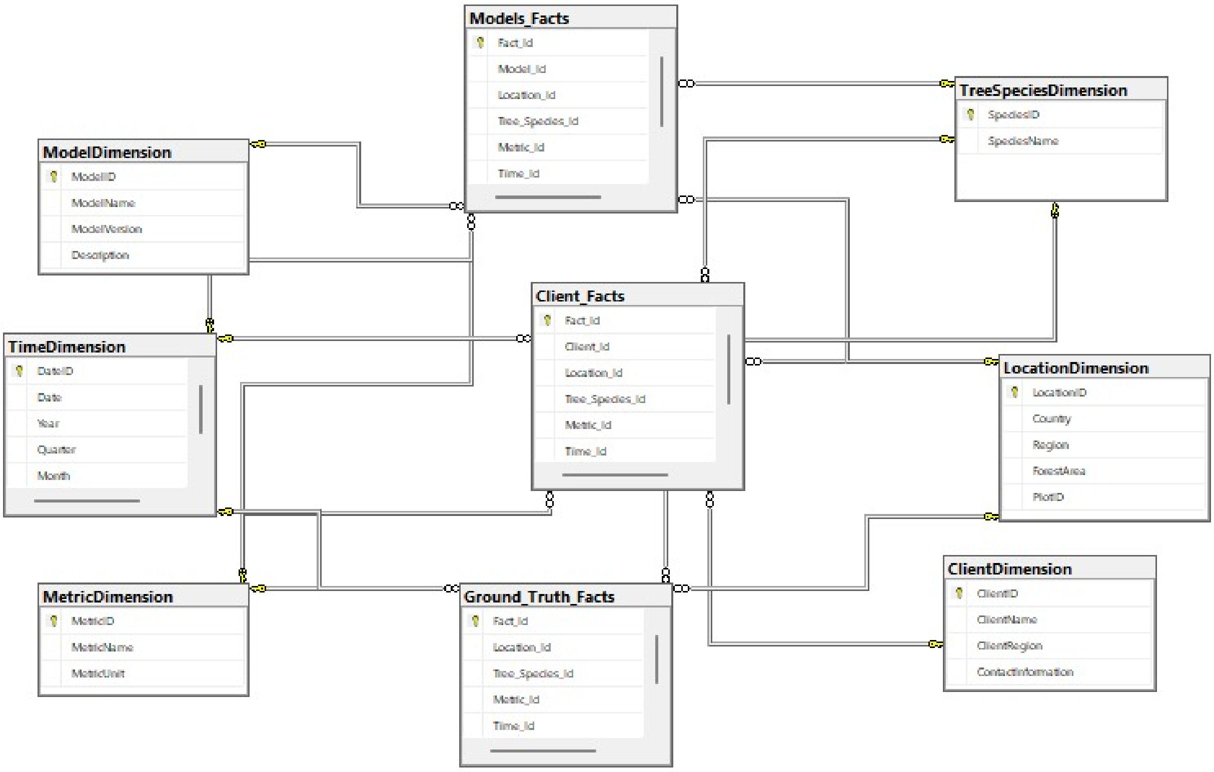


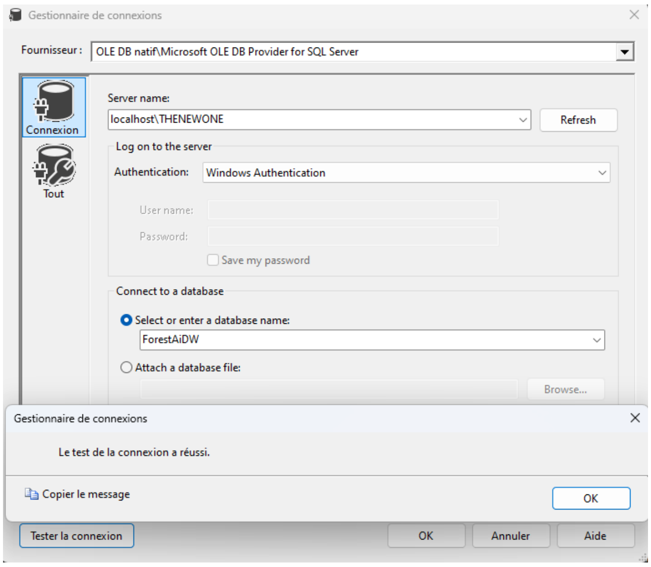
Figure 3.2: Relational Database Schema

3.3 Data Extraction, Transformation, and Loading (ETL)

To manage the ETL process between our relational database and our datamart, we used SQL Server Integration Services (SSIS).

Figure 3.3: SQL Server Integration Services Logo

3.3.1 Connection Manager Establishement



Gestionnaire de connexions

Fournisseur : OLE DB natif\Microsoft OLE DB Provider for SQL Server

Server name: localhost\THENEWONE

Log on to the server

Authentication: Windows Authentication

User name:

Password:

□ Save my password

Connect to a database

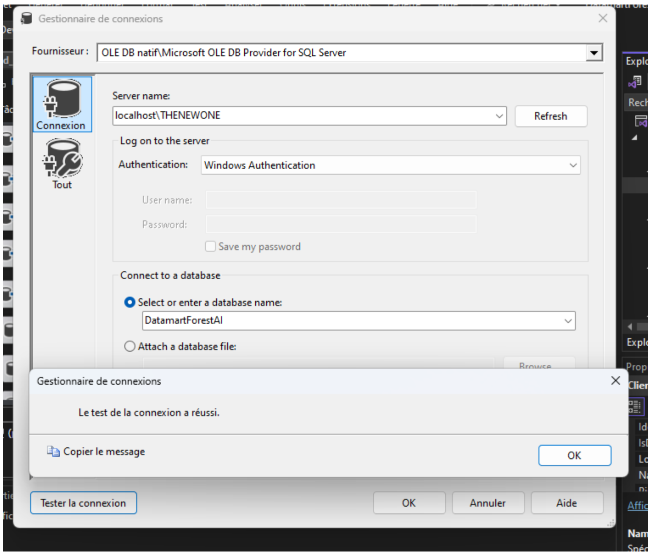
● Select or enter a database name: ForestAiDW

○ Attach a database file:

Gestionnaire de connexions

Le test de la connexion a réussi.

Figure 3.4: Connection to Data Source



Gestionnaire de connexions

Fournisseur : OLE DB natif\Microsoft OLE DB Provider for SQL Server

Server name: localhost\THENEWONE

Log on to the server

Authentication: Windows Authentication

User name:

Password:

□ Save my password

Connect to a database

● Select or enter a database name: DatamartForestAI

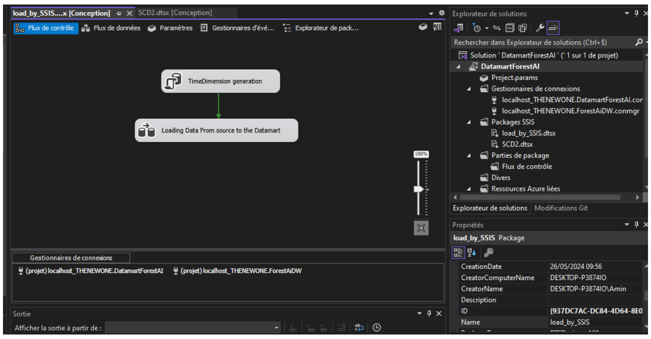
○ Attach a database file:

Gestionnaire de connexions

Le test de la connexion a réussi.

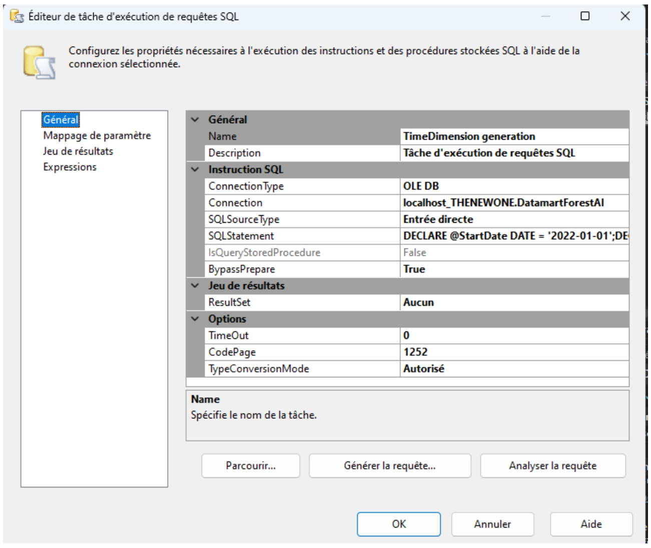
Figure 3.5: Connection to Data Destination

3.3.2 Control Flow and Data Flow Establishment


load_by_SSIS....x [Conception]
SCD2.dtsx [Conception]
Flux de contrôle
Flux de données
Paramètres
Gestionnaires d'évé...
Explorateur de pack...
TimeDimension generation
Loading Data From source to the Datamart
100%
Gestionnaires de connexions
(projet) localhost_THENEWONE.DatamartForestAI
(projet) localhost_THENEWONE.ForestAiDW
Sortie
Afficher la sortie à partir de :
Explorateur de solutions
Rechercher dans Explorateur de solutions (Ctrl+$)
Solution ' DatamartForestAI ' (' 1 sur 1 de projet)
DatamartForestAI
Project.params
Gestionnaires de connexions
localhost_THENEWONE.DatamartForestAI.cor
localhost_THENEWONE.ForestAiDW.conmgr
Packages SSIS
load_by_SSIS.dtsx
SCD2.dtsx
Parties de package
Flux de contrôle
Divers
Ressources Azure liées
Explorateur de solutions
Modifications Git
Propriétés
load_by_SSIS Package

CreationDate	26/05/2024 09:56
CreatorComputerName	DESKTOP-P3874IO
CreatorName	DESKTOP-P3874IO\Amin
Description	
ID	{937DC7AC-DC84-4D64-8E0
Name	load_by_SSIS

Figure 3.6: Control Flow Tasks

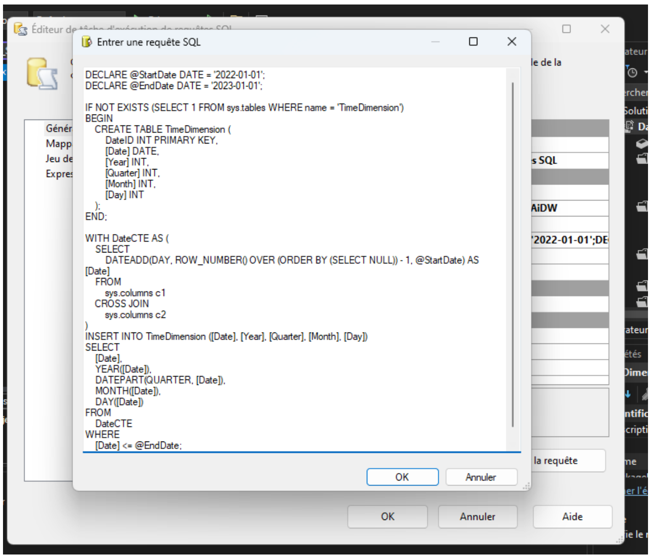


Configurez les propriétés nécessaires à l'exécution des instructions et des procédures stockées SQL à l'aide de la connexion sélectionnée.

Général	
Name	TimeDimension generation
Description	Tâche d'exécution de requêtes SQL
Instruction SQL	
ConnectionType	OLE DB
Connection	localhost_THENEWONE.DatamartForestAI
SQLSourceType	Entrée directe
SQLStatement	DECLARE @StartDate DATE = '2022-01-01';DE
IsQueryStoredProcedure	False
BypassPrepare	True
Jeu de résultats	
ResultSet	Aucun
Options	
TimeOut	0
CodePage	1252
TypeConversionMode	Autorisé

Name
Spécifie le nom de la tâche.

Figure 3.7: Generation of Date Dimension using SSIS



```
DECLARE @StartDate DATE = '2022-01-01';
DECLARE @EndDate DATE = '2023-01-01';

IF NOT EXISTS (SELECT 1 FROM sys.tables WHERE name = 'TimeDimension')
BEGIN
   CREATE TABLE TimeDimension (
      DateID INT PRIMARY KEY,
      [Date] DATE,
      [Year] INT,
      [Quarter] INT,
      [Month] INT,
      [Day] INT
   );
END;

WITH DateCTE AS (
   SELECT
      DATEADD(DAY, ROW_NUMBER() OVER (ORDER BY (SELECT NULL)) - 1, @StartDate) AS
[Date]
   FROM
      sys.columns c1
   CROSS JOIN
      sys.columns c2
)
INSERT INTO TimeDimension ([Date], [Year], [Quarter], [Month], [Day])
SELECT
   [Date],
   YEAR([Date]),
   DATEPART(QUARTER, [Date]),
   MONTH([Date]),
   DAY([Date])
FROM
   DateCTE
WHERE
   [Date] <= @EndDate;
```

Figure 3.8: SQL code to generate the Date Dimension.

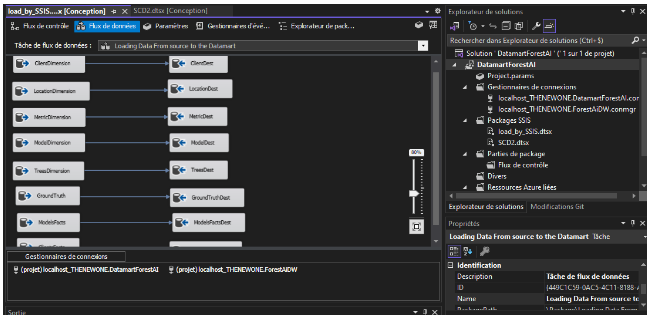
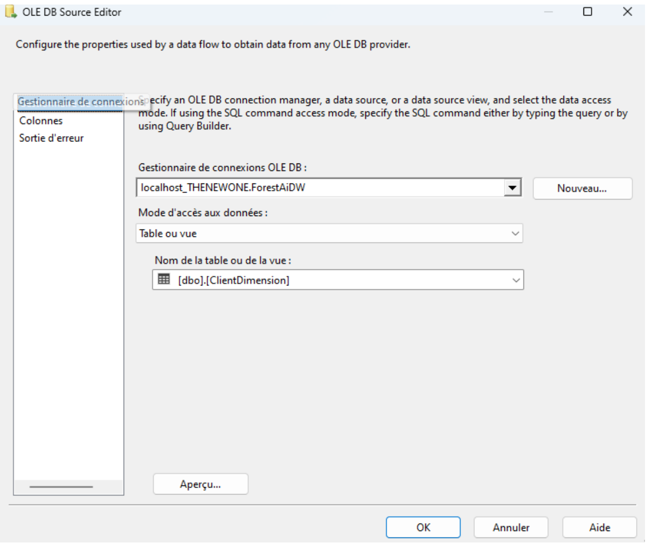


Figure 3.9: Data Flow Tasks



Configure the properties used by a data flow to obtain data from any OLE DB provider.

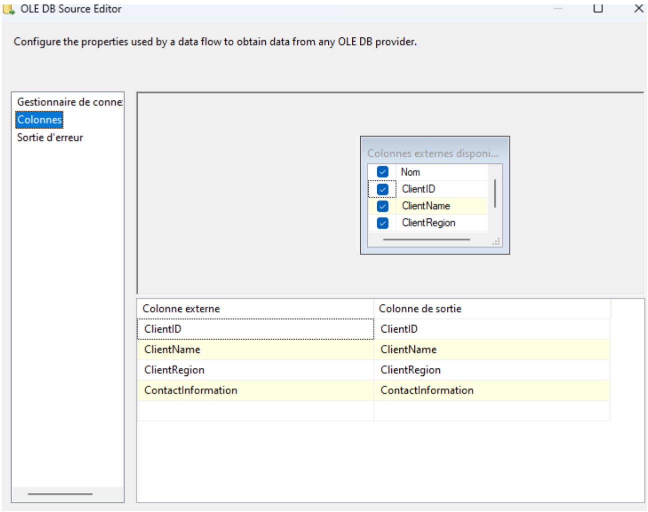
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

Gestionnaire de connexions OLE DB : localhost_THENEWONE.ForestAiDW

Mode d'accès aux données : Table ou vue

Nom de la table ou de la vue : [dbo].[ClientDimension]

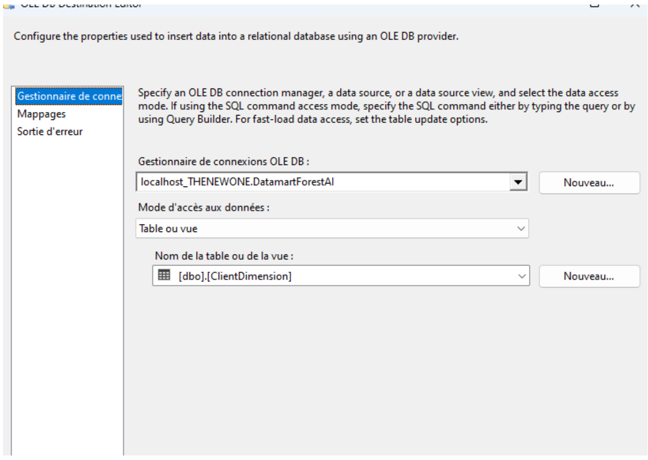
Figure 3.10: Data Source Configuration


OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Colonne externe	Colonne de sortie
ClientID	ClientID
ClientName	ClientName
ClientRegion	ClientRegion
ContactInformation	ContactInformation

Figure 3.11: Data Source Columns Configuration


Configure the properties used to insert data into a relational database using an OLE DB provider.

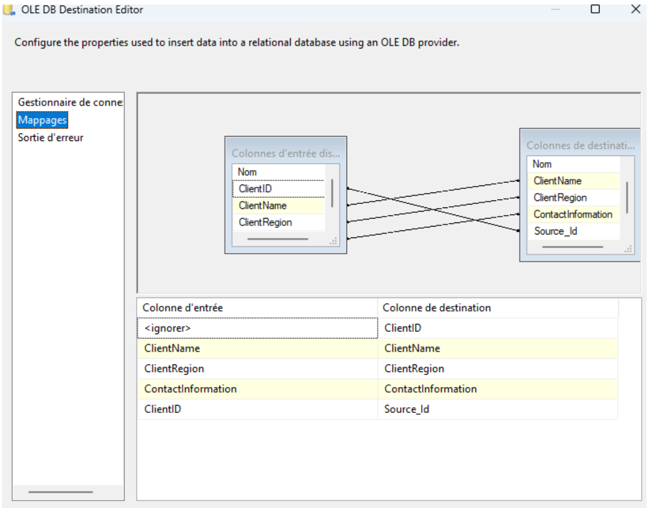
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.

Gestionnaire de connexions OLE DB : localhost_THENEWONE.DatamartForestAI

Mode d'accès aux données : Table ou vue

Nom de la table ou de la vue : [dbo].[ClientDimension]

Figure 3.12: Data Destination Configuration


OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

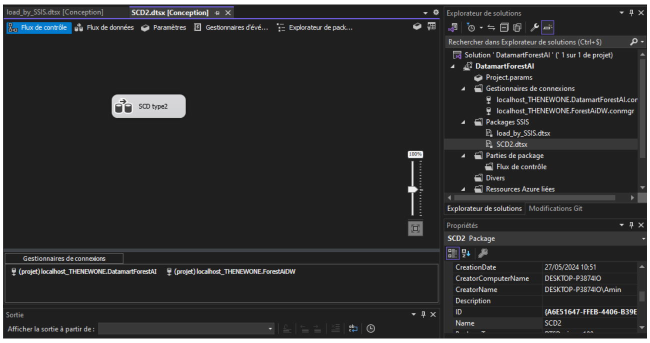
Colonne d'entrée	Colonne de destination
<ignorer>	ClientID
ClientName	ClientName
ClientRegion	ClientRegion
ContactInformation	ContactInformation
ClientID	Source_Id

Figure 3.13: Mapping between Data Source columns and Data Destination columns

This is an example configuration for ETL data from our source to the Client Dimension, we proceeded with the same steps to populate all the rest of the tables in our data mart.

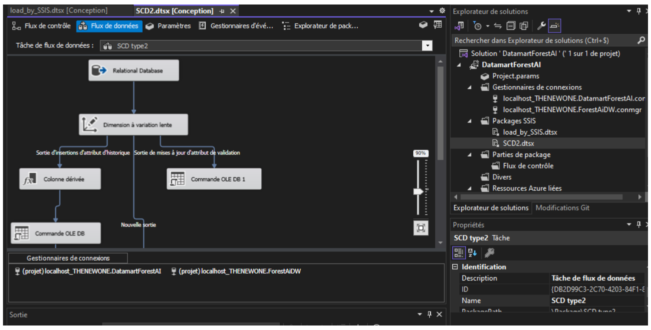
3.4 Slowly Changing Dimensions (SCDs)

We have applied Slowly Changing Dimension type 2 on the Model Dimension.We choose the model version as the historical data, and we added a column "IsCurrent" which can take two values : true and false, to keep track of the new and old data.


load_by_SSIS.dtsx [Conception]
SCD2.dtsx [Conception]
Flux de contrôle
Flux de données
Paramètres
Gestionnaires d'évé...
Explorateur de pack...
SCD type2
100%
Gestionnaires de connexions
(projet) localhost_THENEWONE.DatamartForestAI
(projet) localhost_THENEWONE.ForestAiDW
Sortie
Afficher la sortie à partir de :
Explorateur de solutions
Rechercher dans Explorateur de solutions (Ctrl+$)
Solution ' DatamartForestAI ' (' 1 sur 1 de projet)
DatamartForestAI
Project.params
Gestionnaires de connexions
localhost_THENEWONE.DatamartForestAI.cor
localhost_THENEWONE.ForestAiDW.conmgr
Packages SSIS
load_by_SSIS.dtsx
SCD2.dtsx
Parties de package
Flux de contrôle
Divers
Ressources Azure liées
Explorateur de solutions
Modifications Git
Propriétés
SCD2 Package

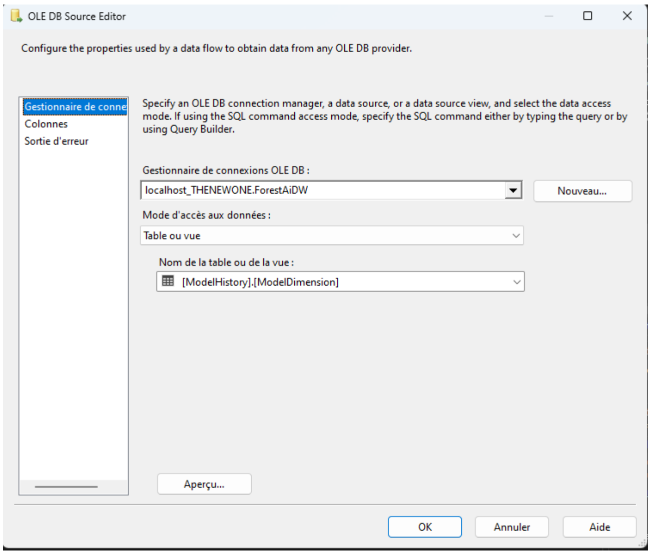
CreationDate	27/05/2024 10:51
CreatorComputerName	DESKTOP-P3874IO
CreatorName	DESKTOP-P3874IO\Amin
Description	
ID	{A6E51647-FFEB-4406-B39E
Name	SCD2

Figure 3.14: Control Flow Task


load_by_SSIS.dtsx [Conception]
SCD2.dtsx [Conception]
Flux de contrôle
Flux de données
Paramètres
Gestionnaires d'évé...
Explorateur de pack...
Tâche de flux de données :
SCD type2
Relational Database
Dimension à variation lente
Sortie d'insertions d'attribut d'historique
Sortie de mises à jour d'attribut de validation
Colonne dérivée
Commande OLE DB 1
Commande OLE DB
Nouvelle sortie
90%
Gestionnaires de connexions
(projet) localhost_THENEWONE.DatamartForestAI
(projet) localhost_THENEWONE.ForestAiDW
Sortie
Explorateur de solutions
Rechercher dans Explorateur de solutions (Ctrl+$)
Solution ' DatamartForestAI ' (' 1 sur 1 de projet)
DatamartForestAI
Project.params
Gestionnaires de connexions
localhost_THENEWONE.DatamartForestAI.cor
localhost_THENEWONE.ForestAiDW.conmgr
Packages SSIS
load_by_SSIS.dtsx
SCD2.dtsx
Parties de package
Flux de contrôle
Divers
Ressources Azure liées
Explorateur de solutions
Modifications Git
Propriétés
SCD type2 Tâche
Identification

Description	Tâche de flux de données
ID	{DB2D99C3-2C70-4203-84F1-E
Name	SCD type2

Figure 3.15: Data Flow Tasks



Configure the properties used by a data flow to obtain data from any OLE DB provider.

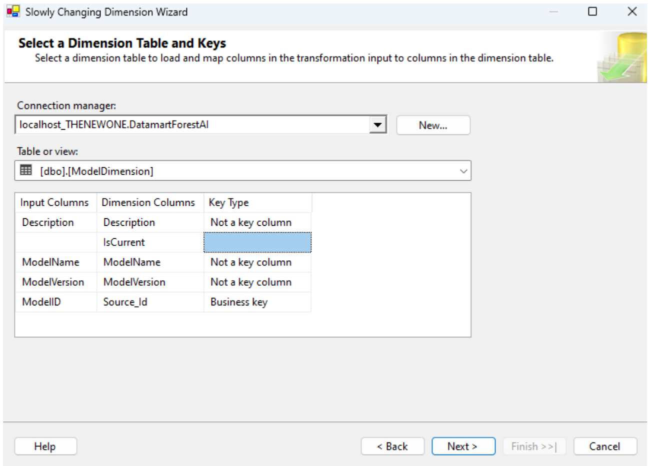
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder.

Gestionnaire de connexions OLE DB : localhost_THENEWONE.ForestAiDW

Mode d'accès aux données : Table ou vue

Nom de la table ou de la vue : [ModelHistory].[ModelDimension]

Figure 3.16: Data Source Specification



Select a Dimension Table and Keys

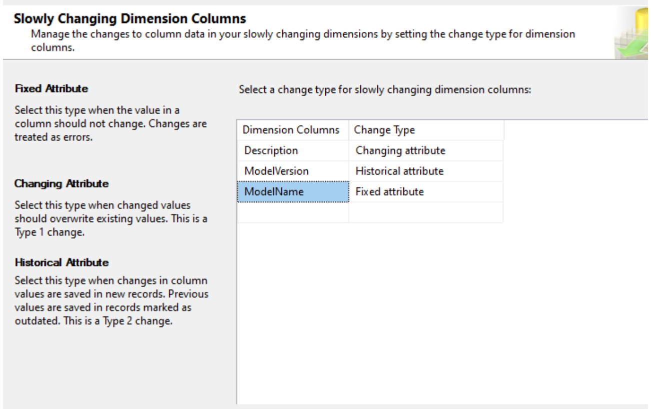
Select a dimension table to load and map columns in the transformation input to columns in the dimension table.

Connection manager: localhost_THENEWONE.DatamartForestAI

Table or view: [dbo].[ModelDimension]

Input Columns	Dimension Columns	Key Type
Description	Description	Not a key column
	IsCurrent	
ModelName	ModelName	Not a key column
ModelVersion	ModelVersion	Not a key column
ModelID	Source_Id	Business key

Figure 3.17: Selection of Dimension Table



Slowly Changing Dimension Columns

Manage the changes to column data in your slowly changing dimensions by setting the change type for dimension columns.

Fixed Attribute

Select this type when the value in a column should not change. Changes are treated as errors.

Changing Attribute

Select this type when changed values should overwrite existing values. This is a Type 1 change.

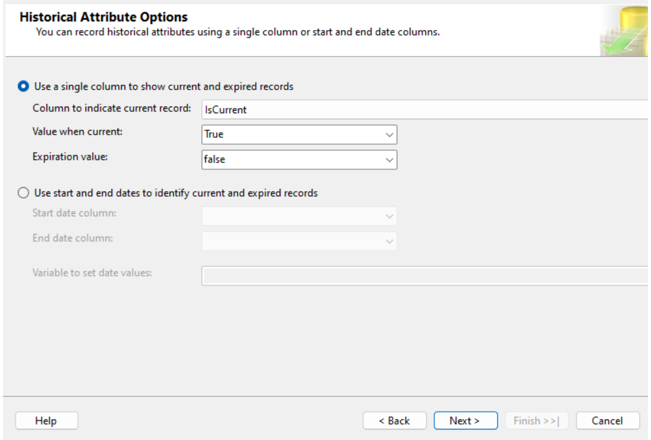
Historical Attribute

Select this type when changes in column values are saved in new records. Previous values are saved in records marked as outdated. This is a Type 2 change.

Select a change type for slowly changing dimension columns:

Dimension Columns	Change Type
Description	Changing attribute
ModelVersion	Historical attribute
ModelName	Fixed attribute

Figure 3.18: Selection of "Change" Type for Slowely Changing Dimension Columns



Historical Attribute Options

You can record historical attributes using a single column or start and end date columns.

● Use a single column to show current and expired records

Column to indicate current record: IsCurrent

Value when current: True

Expiration value: false

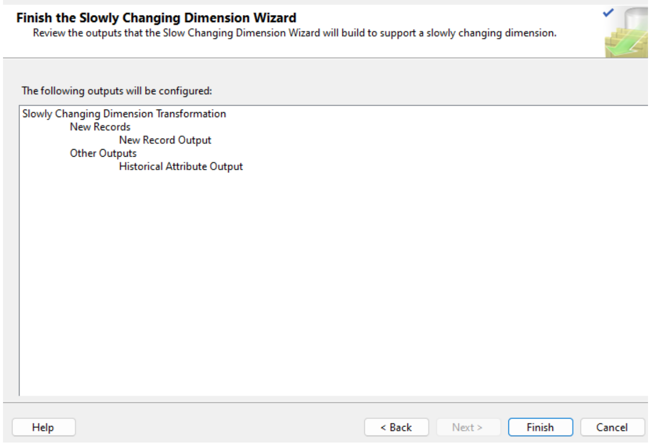
○ Use start and end dates to identify current and expired records

Start date column:

End date column:

Variable to set date values:

Figure 3.19: Use of a Single Columns to Show Current and Expired Records



Finish the Slowly Changing Dimension Wizard

Review the outputs that the Slow Changing Dimension Wizard will build to support a slowly changing dimension.

The following outputs will be configured:

```
Slowly Changing Dimension Transformation
        New Records
                New Record Output
        Other Outputs
                Historical Attribute Output
```

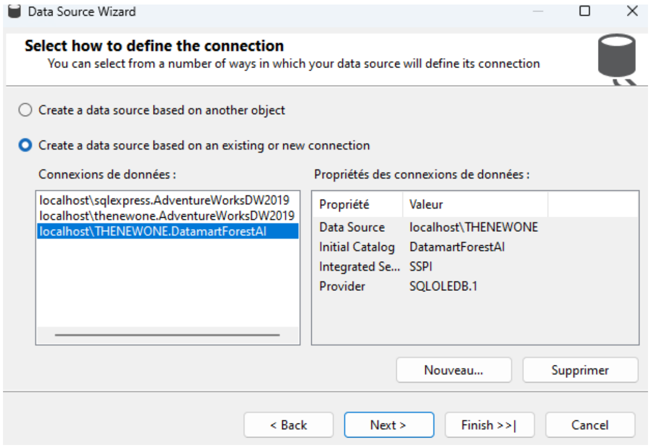
Figure 3.20: Finish the Slowely Changing Dimension Configuration

3.5 Cube Construction

In constructing our cube, we leveraged the capabilities of SSAS (SQL Server Analysis Services). This powerful tool allowed us to model and organize our data efficiently, facilitating insightful analysis and reporting.

Figure 3.21: SQL Server Analysis Services

3.5.1 Data Source Configuration



Select how to define the connection

You can select from a number of ways in which your data source will define its connection

○ Create a data source based on another object

● Create a data source based on an existing or new connection

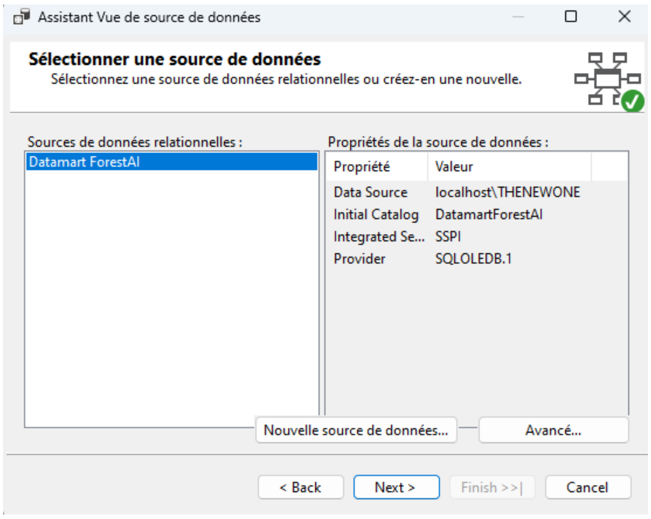
Connexions de données :

localhost\sqlexpress.AdventureWorksDW2019
localhost\thenewone.AdventureWorksDW2019
localhost\THENEWONE.DatamartForestAI

Propriétés des connexions de données :

Propriété	Valeur
Data Source	localhost\THENEWONE
Initial Catalog	DatamartForestAI
Integrated Se...	SSPI
Provider	SQLOLEDB.1

Figure 3.22: Selection of Connexion Type



Assistant Vue de source de données

Sélectionner une source de données

Sélectionnez une source de données relationnelles ou créez-en une nouvelle.

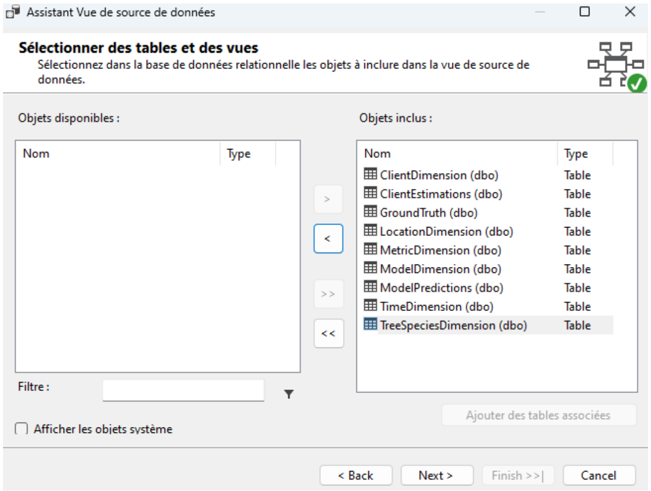
Sources de données relationnelles :

Datamart ForestAI

Propriétés de la source de données :

Propriété	Valeur
Data Source	localhost\THENEWONE
Initial Catalog	DatamartForestAI
Integrated Se...	SSPI
Provider	SQLOLEDB.1

Figure 3.23: Data Source Configuration-Connexion



Assistant Vue de source de données

Sélectionner des tables et des vues

Sélectionnez dans la base de données relationnelle les objets à inclure dans la vue de source de données.

Objets disponibles :

Nom	Type

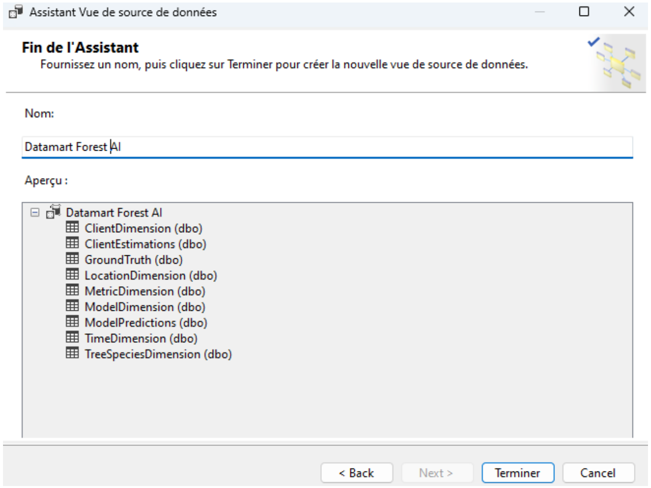
Objets inclus :

Nom	Type
ClientDimension (dbo)	Table
ClientEstimations (dbo)	Table
GroundTruth (dbo)	Table
LocationDimension (dbo)	Table
MetricDimension (dbo)	Table
ModelDimension (dbo)	Table
ModelPredictions (dbo)	Table
TimeDimension (dbo)	Table
TreeSpeciesDimension (dbo)	Table

Filtre :

□ Afficher les objets système

Figure 3.24: Data Source Configuration-Tables Selection



Assistant Vue de source de données

Fin de l'Assistant

Fournissez un nom, puis cliquez sur Terminer pour créer la nouvelle vue de source de données.

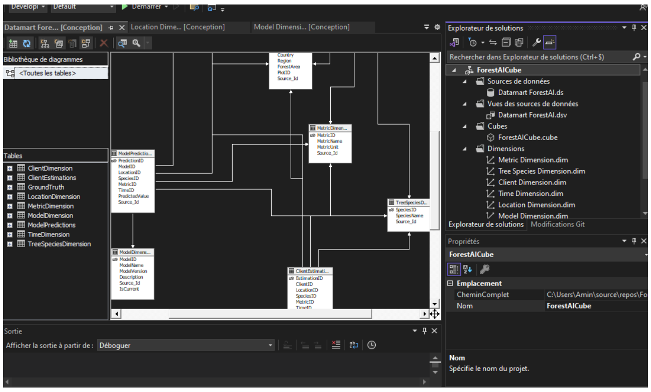
Nom:

Datamart Forest AI

Aperçu :

- Datamart Forest AI
  - ClientDimension (dbo)
  - ClientEstimations (dbo)
  - GroundTruth (dbo)
  - LocationDimension (dbo)
  - MetricDimension (dbo)
  - ModelDimension (dbo)
  - ModelPredictions (dbo)
  - TimeDimension (dbo)
  - TreeSpeciesDimension (dbo)

Figure 3.25: Data Source Configuration-Result



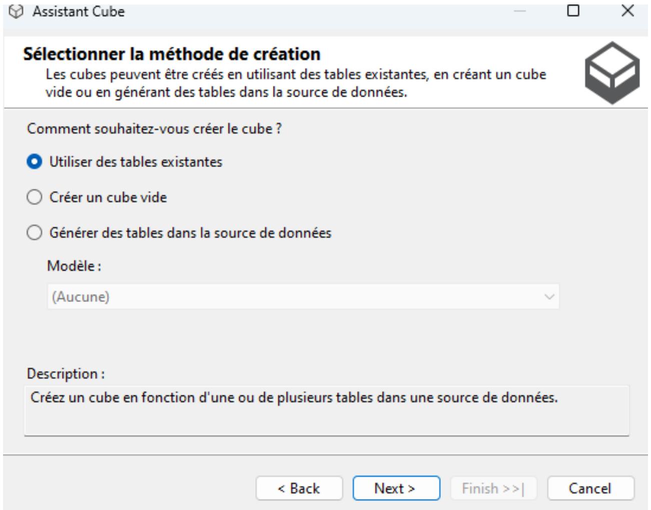
CheminComplet	C:\Users\Amin\source\repos\Fo
Nom	ForestAICube

Nom

Spécifie le nom du projet.

Figure 3.26: Data Source Configuration-Result Shema

3.5.2 Cube Construction



Assistant Cube

Sélectionner la méthode de création

Les cubes peuvent être créés en utilisant des tables existantes, en créant un cube vide ou en générant des tables dans la source de données.

Comment souhaitez-vous créer le cube ?

● Utiliser des tables existantes

○ Créer un cube vide

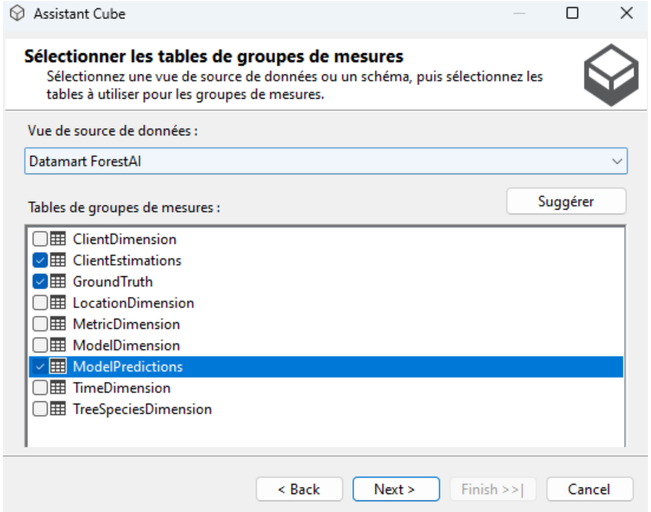
○ Générer des tables dans la source de données

Modèle : (Aucune)

Description :

Créez un cube en fonction d'une ou de plusieurs tables dans une source de données.

Figure 3.27: Cube Construction-Selection of Creation Method



Assistant Cube

Sélectionner les tables de groupes de mesures

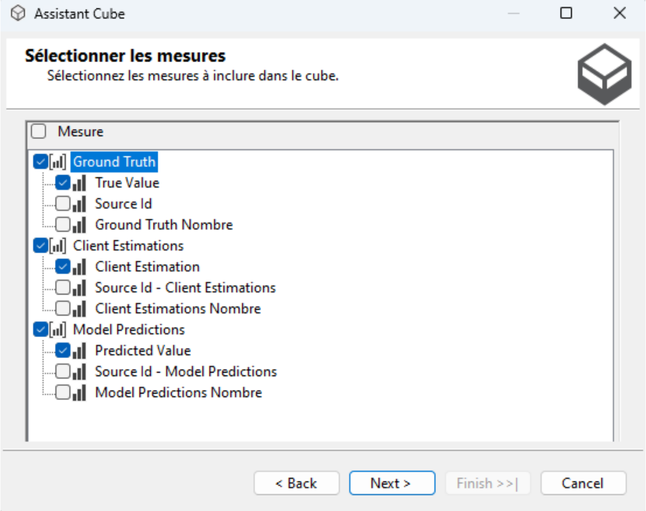
Sélectionnez une vue de source de données ou un schéma, puis sélectionnez les tables à utiliser pour les groupes de mesures.

Vue de source de données : Datamart ForestAI

Tables de groupes de mesures :

- □ ClientDimension
- ☑ ClientEstimations
- ☑ GroundTruth
- □ LocationDimension
- □ MetricDimension
- □ ModelDimension
- ☑ ModelPredictions
- □ TimeDimension
- □ TreeSpeciesDimension

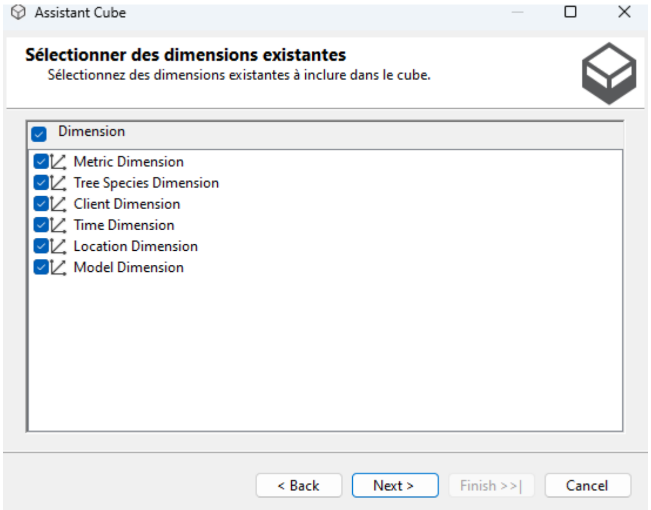
Figure 3.28: Cube Construction-Selection of Facts Tables



Sélectionner les mesures
Sélectionnez les mesures à inclure dans le cube.

- ☐ Mesure
- ☑ Ground Truth
  - ☑ True Value
  - ☐ Source Id
  - ☐ Ground Truth Nombre
- ☑ Client Estimations
  - ☑ Client Estimation
  - ☐ Source Id - Client Estimations
  - ☐ Client Estimations Nombre
- ☑ Model Predictions
  - ☑ Predicted Value
  - ☐ Source Id - Model Predictions
  - ☐ Model Predictions Nombre

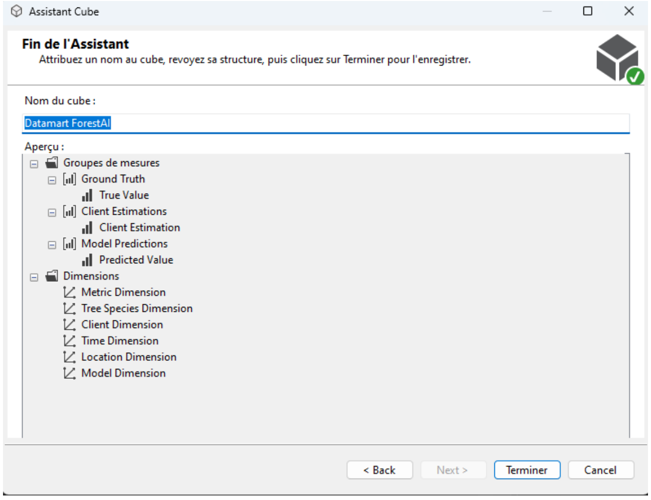
Figure 3.29: Cube Construction-Selection of Metrics



Sélectionner des dimensions existantes
Sélectionnez des dimensions existantes à inclure dans le cube.

- ☑ Dimension
- ☑ Metric Dimension
- ☑ Tree Species Dimension
- ☑ Client Dimension
- ☑ Time Dimension
- ☑ Location Dimension
- ☑ Model Dimension

Figure 3.30: Cube Construction-Selection of Dimensions



Fin de l'Assistant

Attribuez un nom au cube, revoyez sa structure, puis cliquez sur Terminer pour l'enregistrer.

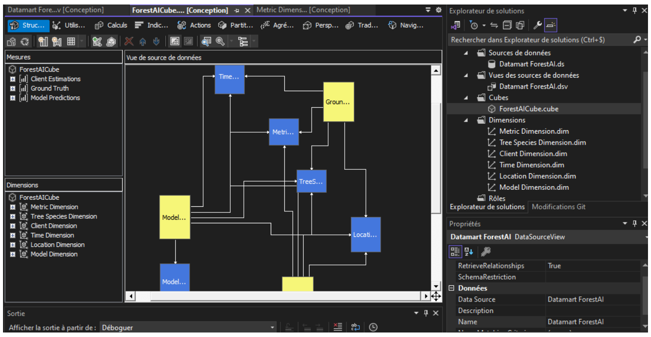
Nom du cube :

Datamart ForestAI

Aperçu :

- Groupes de mesures
  - Ground Truth
    - True Value
  - Client Estimations
    - Client Estimation
  - Model Predictions
    - Predicted Value
- Dimensions
  - Metric Dimension
  - Tree Species Dimension
  - Client Dimension
  - Time Dimension
  - Location Dimension
  - Model Dimension

Figure 3.31: Cube Construction-Finsh the Construction



Mesures
- ForestAICube
  - Client Estimations
  - Ground Truth
  - Model Predictions

Dimensions
- ForestAICube
  - Metric Dimension
  - Tree Species Dimension
  - Client Dimension
  - Time Dimension
  - Location Dimension
  - Model Dimension

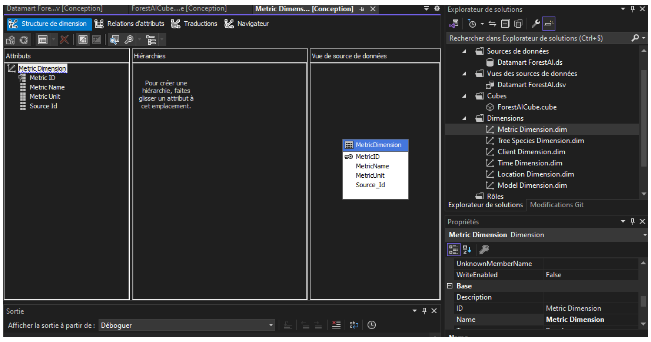
Afficher la sortie à partir de : Déboguer

- Sources de données
  - Datamart ForestAI.ds
- Vues des sources de données
  - Datamart ForestAI.dsv
- Cubes
  - ForestAICube.cube
- Dimensions
  - Metric Dimension.dim
  - Tree Species Dimension.dim
  - Client Dimension.dim
  - Time Dimension.dim
  - Location Dimension.dim
  - Model Dimension.dim
- Rôles

Datamart ForestAI DataSourceView

RetrieveRelationships	True
SchemaRestriction	
Données	
Data Source	Datamart ForestAI
Description	
Name	Datamart ForestAI

Figure 3.32: Cube Construction-Result



Attributs
- Metric Dimension
  - Metric ID
  - Metric Name
  - Metric Unit
  - Source Id

Pour créer une hiérarchie, faites glisser un attribut à cet emplacement.

MetricDimension
- MetricID
- MetricName
- MetricUnit
- Source_Id

Afficher la sortie à partir de : Déboguer

- Sources de données
  - Datamart ForestAI.ds
- Vues des sources de données
  - Datamart ForestAI.dsv
- Cubes
  - ForestAICube.cube
- Dimensions
  - Metric Dimension.dim
  - Tree Species Dimension.dim
  - Client Dimension.dim
  - Time Dimension.dim
  - Location Dimension.dim
  - Model Dimension.dim
- Rôles

Metric Dimension Dimension

UnknownMemberName	
WriteEnabled	False
Base	
Description	
ID	Metric Dimension
Name	**Metric Dimension**

Figure 3.33: Attributes of metrics used in the cube - Example1

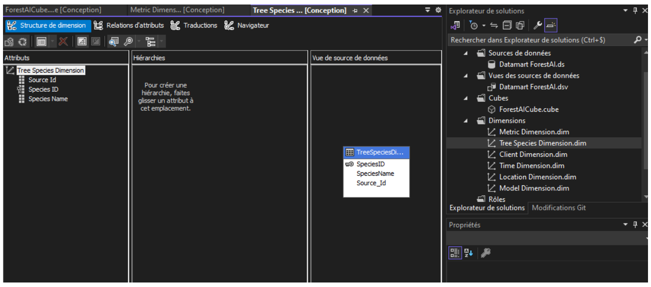
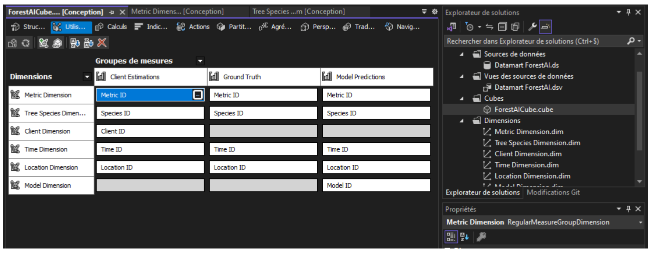


Figure 3.34: Attributes of metrics used in the cube - Example2

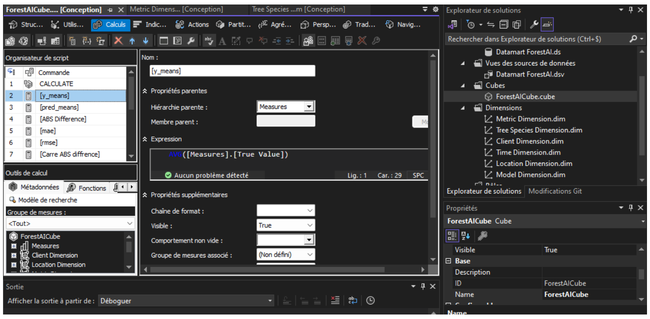


Groupes de mesures

Dimensions	Client Estimations	Ground Truth	Model Predictions
Metric Dimension	Metric ID	Metric ID	Metric ID
Tree Species Dimen...	Species ID	Species ID	Species ID
Client Dimension	Client ID		
Time Dimension	Time ID	Time ID	Time ID
Location Dimension	Location ID	Location ID	Location ID
Model Dimension			Model ID

ForestAICube.... [Conception]
Metric Dimens... [Conception]
Tree Species ...m [Conception]
Struc...
Utilis...
Calculs
Indic...
Actions
Partit...
Agré...
Persp...
Trad...
Navig...
Explorateur de solutions
Propriétés
Metric Dimension RegularMeasureGroupDimension

Figure 3.35: Cube Granularity



Organisateur de script

	Commande
1	CALCULATE
2	[y_means]
3	[pred_means]
4	[ABS Difference]
5	[mae]
6	[rmse]
7	[Carre ABS diffrence]

Nom : [y_means]

Propriétés parentes

Hiérarchie parente : Measures

Membre parent :

Expression

```
AVG([Measures].[True Value])
```

Aucun problème détecté Lig. : 1 Car. : 29 SPC

Propriétés supplémentaires

Chaîne de format :

Visible : True

Comportement non vide :

Groupe de mesures associé : (Non défini)

Figure 3.36: Adding Calculable Facts to the Cube

3.6 Conclusion

In conclusion, our data warehouse implementation was executed successfully. We established a critical relational database using Microsoft SQL Server Management Studio, then effectively designed and implemented the ETL process to ensure data integrity. Using SSAS, we constructed a cube with essential dimensions, facts, and aggregated measures, resulting in a robust and efficient analytical environment ready for comprehensive data analysis and visualization.

CHAPTER 4

DATA ANALYSIS AND VISUALIZATION

4.1 Introduction

The visualization phase of our data warehouse, achieved through Power BI connected to our cube, represents a critical step in interpreting and effectively communicating our data. This section highlights the different reports and visualizations created to analyze model performance, validate results, and compare predictions with real data. By combining the power of our data cube with the interactive features of Power BI, we have developed highly informative visual tools to guide decision-making and communicate our findings clearly and persuasively.

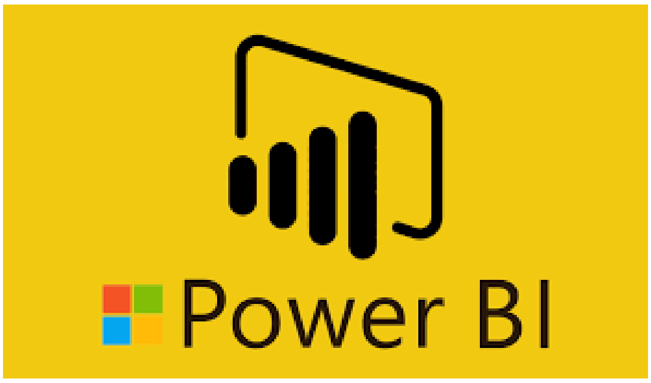
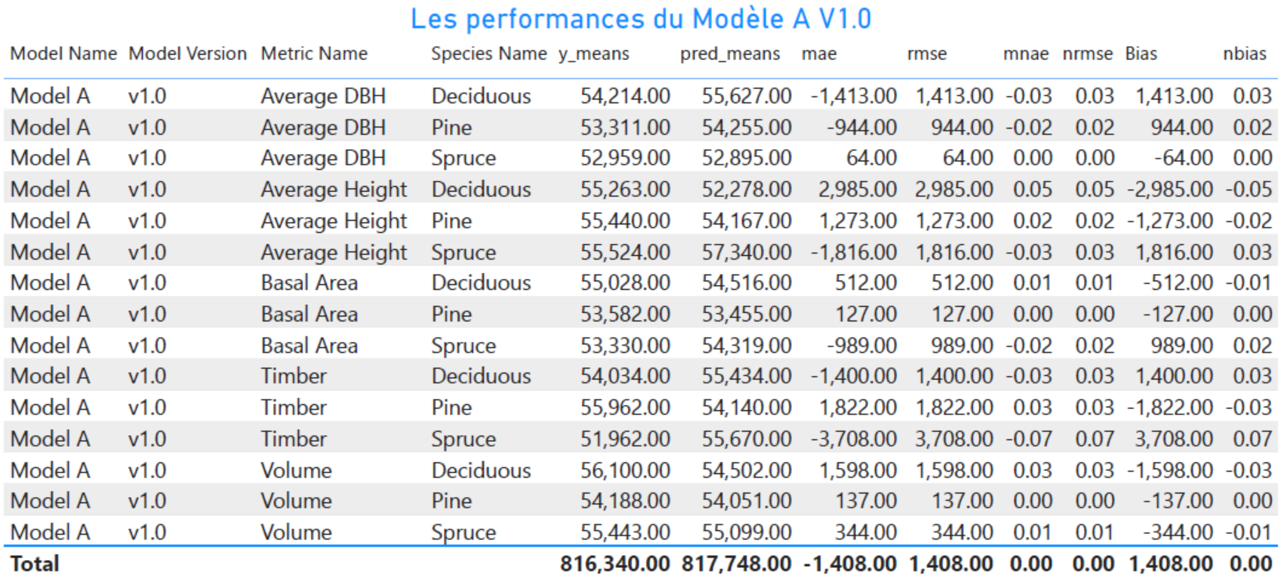


Figure 4.1: Microsoft PowerBI Logo

4.2 Model Performance Report

Model Performance Reports present characteristics and error metrics for the version of a model.



Les performances du Modèle A V1.0

Model Name	Model Version	Metric Name	Species Name	y_means	pred_means	mae	rmse	mnae	nrmse	Bias	nbias
Model A	v1.0	Average DBH	Deciduous	54,214.00	55,627.00	-1,413.00	1,413.00	-0.03	0.03	1,413.00	0.03
Model A	v1.0	Average DBH	Pine	53,311.00	54,255.00	-944.00	944.00	-0.02	0.02	944.00	0.02
Model A	v1.0	Average DBH	Spruce	52,959.00	52,895.00	64.00	64.00	0.00	0.00	-64.00	0.00
Model A	v1.0	Average Height	Deciduous	55,263.00	52,278.00	2,985.00	2,985.00	0.05	0.05	-2,985.00	-0.05
Model A	v1.0	Average Height	Pine	55,440.00	54,167.00	1,273.00	1,273.00	0.02	0.02	-1,273.00	-0.02
Model A	v1.0	Average Height	Spruce	55,524.00	57,340.00	-1,816.00	1,816.00	-0.03	0.03	1,816.00	0.03
Model A	v1.0	Basal Area	Deciduous	55,028.00	54,516.00	512.00	512.00	0.01	0.01	-512.00	-0.01
Model A	v1.0	Basal Area	Pine	53,582.00	53,455.00	127.00	127.00	0.00	0.00	-127.00	0.00
Model A	v1.0	Basal Area	Spruce	53,330.00	54,319.00	-989.00	989.00	-0.02	0.02	989.00	0.02
Model A	v1.0	Timber	Deciduous	54,034.00	55,434.00	-1,400.00	1,400.00	-0.03	0.03	1,400.00	0.03
Model A	v1.0	Timber	Pine	55,962.00	54,140.00	1,822.00	1,822.00	0.03	0.03	-1,822.00	-0.03
Model A	v1.0	Timber	Spruce	51,962.00	55,670.00	-3,708.00	3,708.00	-0.07	0.07	3,708.00	0.07
Model A	v1.0	Volume	Deciduous	56,100.00	54,502.00	1,598.00	1,598.00	0.03	0.03	-1,598.00	-0.03
Model A	v1.0	Volume	Pine	54,188.00	54,051.00	137.00	137.00	0.00	0.00	-137.00	0.00
Model A	v1.0	Volume	Spruce	55,443.00	55,099.00	344.00	344.00	0.01	0.01	-344.00	-0.01
Total				**816,340.00**	**817,748.00**	**-1,408.00**	**1,408.00**	**0.00**	**0.00**	**1,408.00**	**0.00**

Figure 4.2: Model Performance Report

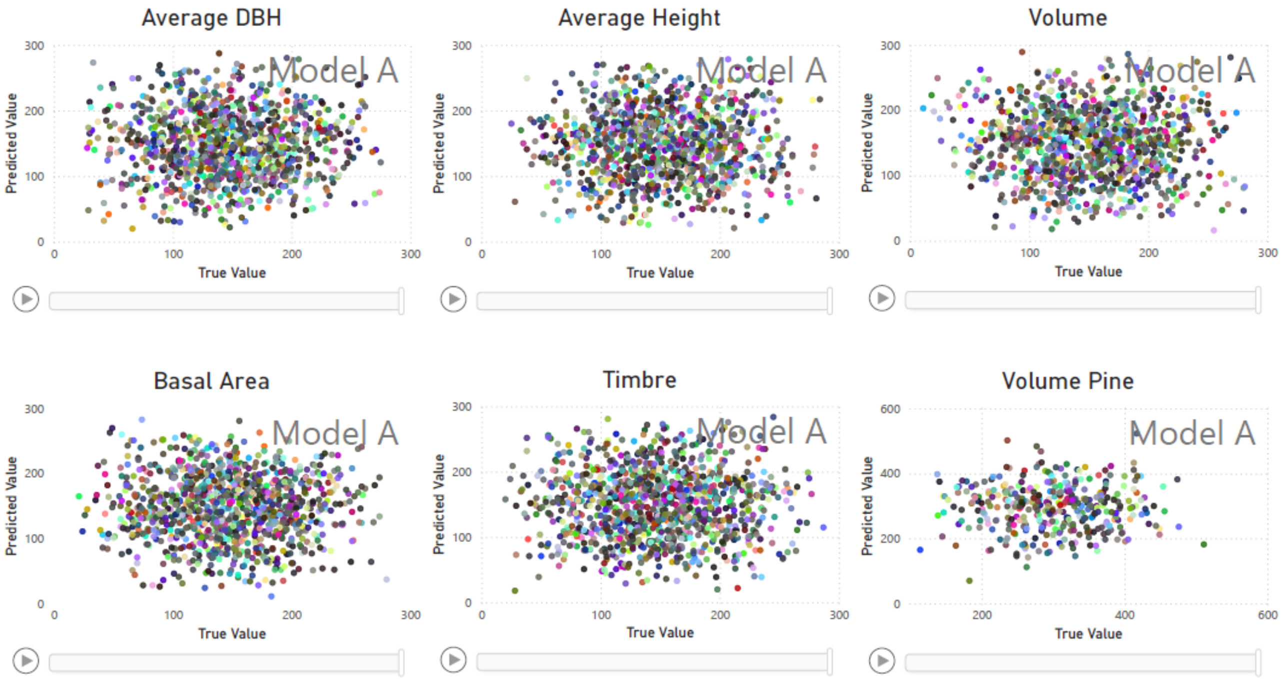


Figure 4.3: Comparison between Real Values and Prediction Values of Facts

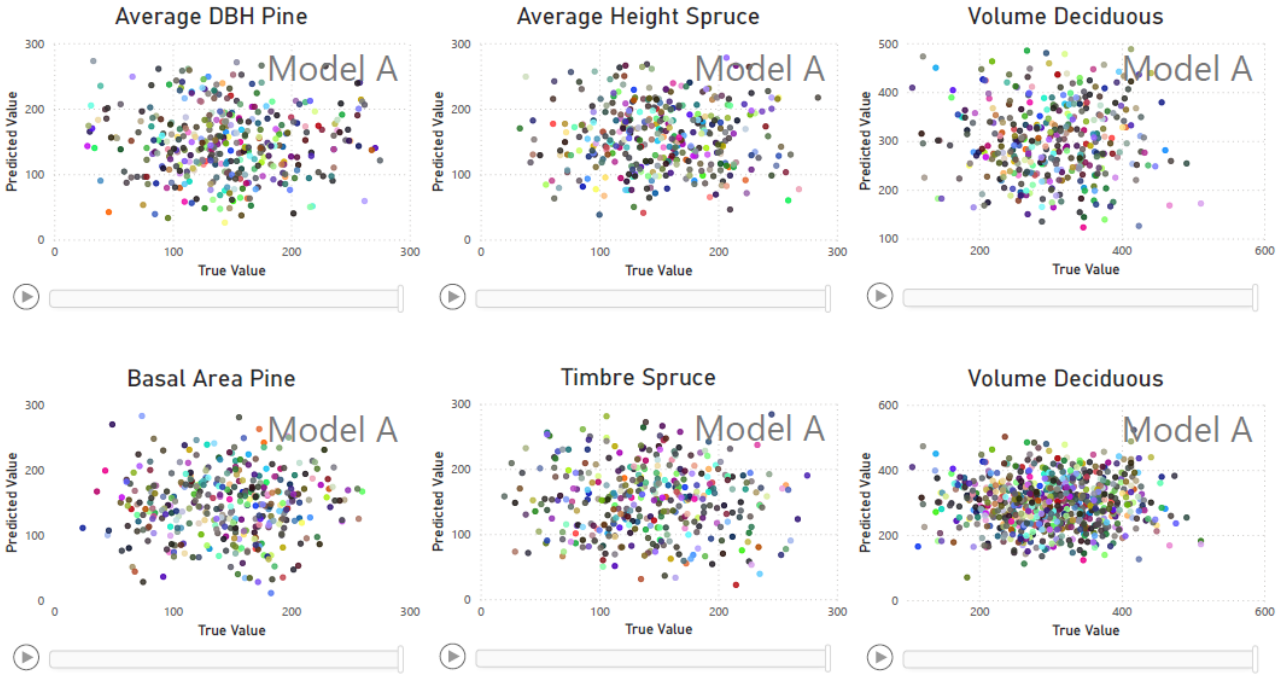
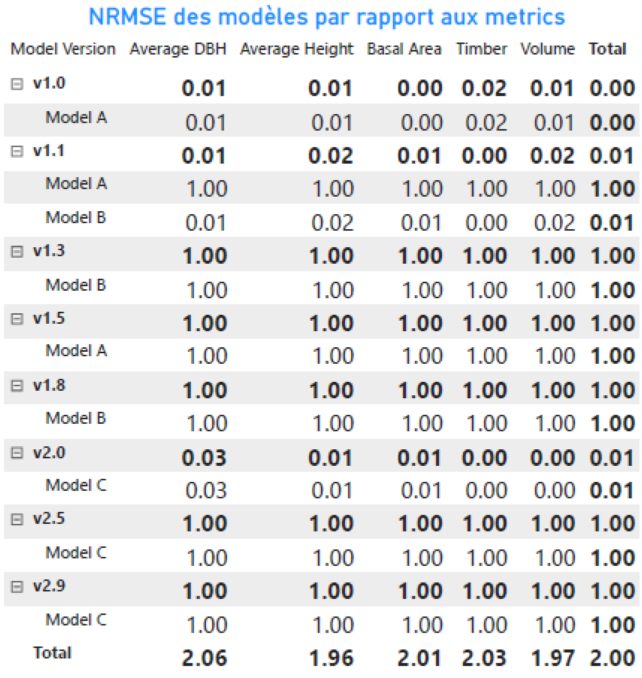


Figure 4.4: Comparison between Real Values and Prediction Values of Facts

4.3 Model Validation Reports

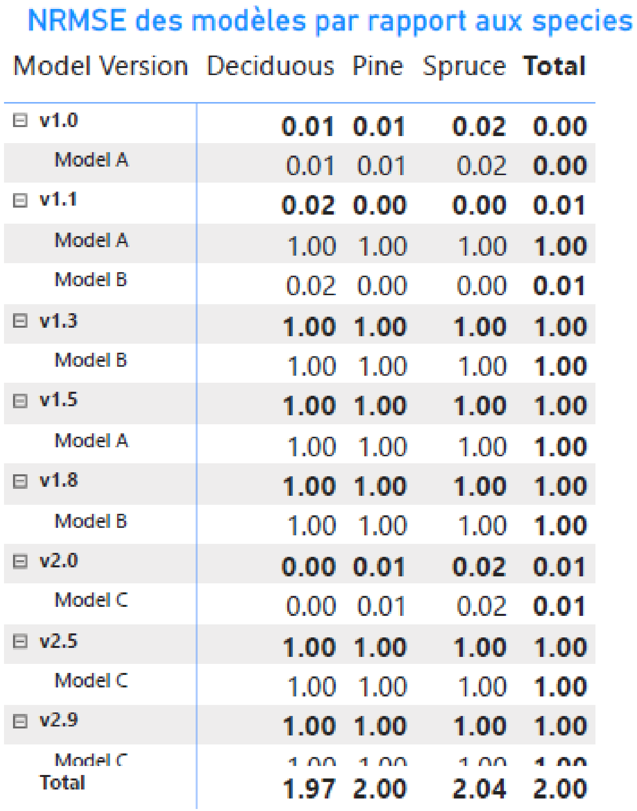
The purpose is to compare the outputs of a new model with those of the previous version to demonstrate its improved performance. This comparison is very necessary to ensure that the new model is functioning as intended and is accurate. The validator evaluates all models against the actual data, using error metrics to identify the model with the most favorable performance.



NRMSE des modèles par rapport aux metrics

Model Version	Average DBH	Average Height	Basal Area	Timber	Volume	**Total**
⊟ **v1.0**	**0.01**	**0.01**	**0.00**	**0.02**	**0.01**	**0.00**
Model A	0.01	0.01	0.00	0.02	0.01	**0.00**
⊟ **v1.1**	**0.01**	**0.02**	**0.01**	**0.00**	**0.02**	**0.01**
Model A	1.00	1.00	1.00	1.00	1.00	**1.00**
Model B	0.01	0.02	0.01	0.00	0.02	**0.01**
⊟ **v1.3**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Model B	1.00	1.00	1.00	1.00	1.00	**1.00**
⊟ **v1.5**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Model A	1.00	1.00	1.00	1.00	1.00	**1.00**
⊟ **v1.8**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Model B	1.00	1.00	1.00	1.00	1.00	**1.00**
⊟ **v2.0**	**0.03**	**0.01**	**0.01**	**0.00**	**0.00**	**0.01**
Model C	0.03	0.01	0.01	0.00	0.00	**0.01**
⊟ **v2.5**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Model C	1.00	1.00	1.00	1.00	1.00	**1.00**
⊟ **v2.9**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Model C	1.00	1.00	1.00	1.00	1.00	**1.00**
Total	**2.06**	**1.96**	**2.01**	**2.03**	**1.97**	**2.00**

Figure 4.5: NRMSE of models relative to metrics



NRMSE des modèles par rapport aux species

Model Version	Deciduous	Pine	Spruce	**Total**
⊟ **v1.0**	**0.01**	**0.01**	**0.02**	**0.00**
Model A	0.01	0.01	0.02	**0.00**
⊟ **v1.1**	**0.02**	**0.00**	**0.00**	**0.01**
Model A	1.00	1.00	1.00	**1.00**
Model B	0.02	0.00	0.00	**0.01**
⊟ **v1.3**	**1.00**	**1.00**	**1.00**	**1.00**
Model B	1.00	1.00	1.00	**1.00**
⊟ **v1.5**	**1.00**	**1.00**	**1.00**	**1.00**
Model A	1.00	1.00	1.00	**1.00**
⊟ **v1.8**	**1.00**	**1.00**	**1.00**	**1.00**
Model B	1.00	1.00	1.00	**1.00**
⊟ **v2.0**	**0.00**	**0.01**	**0.02**	**0.01**
Model C	0.00	0.01	0.02	**0.01**
⊟ **v2.5**	**1.00**	**1.00**	**1.00**	**1.00**
Model C	1.00	1.00	1.00	**1.00**
⊟ **v2.9**	**1.00**	**1.00**	**1.00**	**1.00**
Model C	1.00	1.00	1.00	**1.00**
Total	**1.97**	**2.00**	**2.04**	**2.00**

Figure 4.6: NRMSE of models relative to species

4.4 Client Estimate Comparison Report

This report provides comparisons of the models predictions with client estimations. It is used as a sales tool to convince clients that ForestAI's models are better

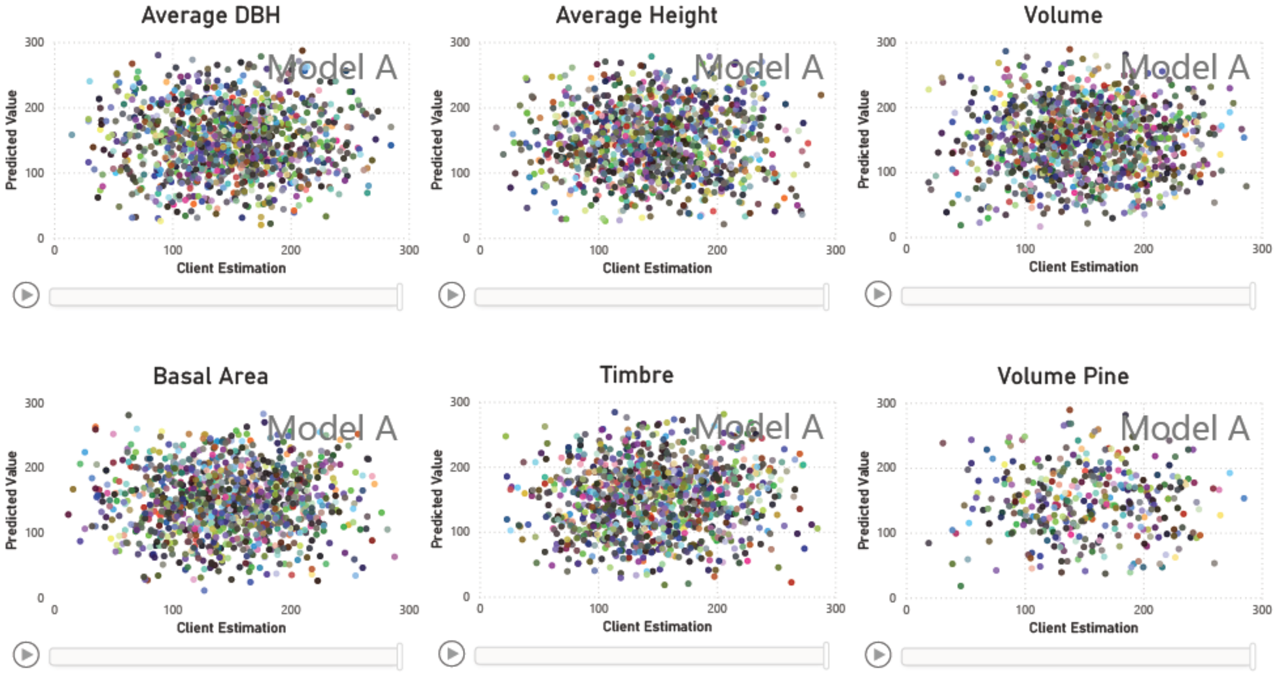


Figure 4.7: Comparison between Client Estimations and Prediction Values of Facts

4.5 Conclusion

Using Power BI as our visualization tool, we have created a range of essential reports to assess our models' performance and delve into our data comprehensively. These reports include comparative analyses of NRMSE relative to various metrics as well as species, providing a comprehensive view of the accuracy and effectiveness of our models. The integration of Power BI with our data cube has enabled us to present this information interactively and convincingly, facilitating informed decision-making and effective communication of results to stakeholders.

SYNTHESIS

In conclusion, the development of the data warehouse for ForestAI has improved model validation, data analysis, and information delivery to clients. Using SSIS for data loading, a robust SSAS cube for detailed analysis, and Power BI for interactive visualizations, the project has achieved its goals of better data management and accessibility.

This technological integration has streamlined the model validation process, allowing for faster and more accurate evaluations, thereby enhancing internal efficiency and client presentations by providing clear, data-driven insights.

By focusing on key metrics and enabling detailed, real-time analysis, ForestAI can better understand model performance and offer superior services. This project demonstrates the crucial role of a well-designed data warehouse in supporting advanced analytics and business intelligence, laying the foundation for continuous growth and operational excellence.

REFERENCES

1. https://www.datacourses.com/evaluation-of-regression-models-in-scikit-learn-846/ (Consulté le 20 mai 2024)
2. https://sefidian.com/2022/08/18/a-guide-on-regression-error-metrics-with-python-code/ (Consulté le 20 mai 2024)
3. https://www.statisticshowto.com/nrmse/ (Consulté le 20 mai 2024)
4. https://www.techtarget.com/searchsoftwarequality/definition/reverse-engineering (Consulté le 27 mai 2024)