НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМ. ІГОРЯ СІКОРСЬКОГО»

ННК «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Системного проектування
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри
_____ А.І. Петренко
(підпис) (ініціали, прізвище)

«___» _____ 2020 р.

Магістерська дисертація

зі спеціальності 122 - комп'ютерні науки
(код і назва спеціальності)

на тему: Аналіз ефективності реалізації алгоритмів Data mining з використанням сервісу Microsoft SQL Server

Виконав (-ла): студент (-ка) 6 курсу, групи ДА-з91мп
(шифр групи)

Ниценко Андрій Сергійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Капшук О.О. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Стартап-проект к.т.н., доцент Капшук О.О. _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент Професор, д.т.н., професор кафедри ММСА _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)
ННК "ІПСА" Бідюк П.І.

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»

Інститут/факультет **Інститут прикладного системного аналізу**
(повна назва)

Кафедра **Системного проектування**
(повна назва)

Рівень вищої освіти – другий (магістерський)

Спеціальність (спеціалізація) **122 Комп'ютерні науки**
(код і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Петренко А.І
(підпис) (ініціали, прізвище)
«__» _____ 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Ниценко Андрій Сергійович
(прізвище, ім'я, по батькові)

1. Тема дисертації «Аналіз ефективності реалізації алгоритмів Data mining з використанням сервісу Microsoft SQL Server»

науковий керівник дисертації Капшук Олег Олексійович, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Строк подання студентом дисертації – 14.12.2020

3. Об'єкт дослідження – алгоритми Data mining в середовищі MS SQL Server і Microsoft Visual Studio

4. Предмет дослідження – ефективність алгоритмів Data mining в середовищі MS SQL Server та Microsoft Visual Studio

5. Перелік завдань, які потрібно виконати:

- Дослідити існуючі алгоритми Data mining в середовищі MS SQL Server і Microsoft Visual Studio;

- Реалізувати алгоритми для задач прогнозування дискретного атрибуту, прогнозування безперервного атрибуту, прогнозування послідовності, знаходження груп спільних елементів в транзакціях;
- Оформити роботу на основі отриманих результатів.

6. Перелік графічного (ілюстративного) матеріалу – презентація на тему «Аналіз ефективності реалізації алгоритмів Data mining з використанням сервісу Microsoft SQL Server»

7. Орієнтовний перелік публікацій _____

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Стартап-проект	Капшук О.О., доцент		

9. Дата видачі завдання 02.09.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з технічною літературою	12.09.2020	
2	Встановлення необхідного ПЗ для дослідження	03.10.2020	
3	Дослідити існуючі алгоритми Data mining в середовищі MS SQL Server і Microsoft Visual Studio	17.10.2020	
4	Реалізувати алгоритми для конкретних задач прогнозування дискретного атрибуту, прогнозування безперервного атрибуту, прогнозування послідовності, знаходження груп спільних елементів в транзакціях	07.11.2020	
5	Оформлення дипломної роботи	13.12.2020	

Студент _____ Ницекно А.С.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Капшук О.О.
(підпис) (ініціали, прізвище)

РЕФЕРАТ
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ

виконану на тему: Аналіз ефективності реалізації алгоритмів Data mining з використанням сервісу Microsoft SQL Server

студентом: Ниценком Андрієм Сергійовичем

Робота виконана на 98 сторінках, містить 61 ілюстрацій, 22 таблиці. При підготовці використовувалася інформація з 33 джерел.

Актуальність теми

На сьогоднішній день даних, які потрібно аналізувати стає все більше і більше. Компанії мають великі обсяги інформації, але мати інформацію замало, потрібно ще й знати якими саме способами її обробляти та аналізувати.

Саме тому дане дослідження має великі перспективи, оскільки розуміння кожного алгоритму Data mining та області його кращого використання буде або спрощувати аналіз для таких компаній, або давати більшу точність отриманих результатів.

Мета та задачі дослідження

Метою даної роботи є реалізація та дослідження ефективності реалізації алгоритмів Data mining, які надає сервіс MS SQL Server та Microsoft Visual Studio, щоб зрозуміти для яких конкретних задач та з яким об’ємом даних буде кращим той чи інший алгоритм.

Рішення поставлених завдань та досягнуті результати

В роботі розглянуто алгоритми (алгоритм кластеризації, алгоритм дерева прийняття рішень, алгоритм лінійної регресії, алгоритм взаємозв'язків, спрощений алгоритм Байеса, алгоритм нейронної мережі, алгоритм кластеризації послідовностей, алгоритм часових рядів), які надає сервіс MS SQL Server та Microsoft Visual Studio, реалізовано кожний з них.

Обрано задачі (прогнозування дискретного атрибуту, прогнозування безперервного атрибуту, прогнозування послідовності, знаходження груп спільних елементів в транзакціях), які потрібно реалізувати та обрано алгоритми відповідно до специфіки завдань. Реалізовано алгоритми Data mining. В ході аналізу оцінки ефективності реалізації алгоритмів було сформовано рекомендації щодо їх використання для різних задач.

Об’єкт досліджень

Алгоритми Data mining в середовищі MS SQL Server і Microsoft Visual Studio.

Предмет досліджень

Ефективність реалізації алгоритмів Data mining в середовищі MS SQL Server і Microsoft Visual Studio.

Методи досліджень

Для розв’язання зазначеної проблеми в роботі проведено порівняння реалізованих алгоритмів Data mining, виділено задачі на основі яких буде проведено аналіз та виділено основні функціональні особливості кожного алгоритму.

Наукова новизна

Наукова новизна полягає в тому, що після реалізації кожного алгоритму для конкретної задачі встановлено, який саме алгоритм краще підходить для кожної задачі, щоб в майбутньому малі компанії, які будуть займатися аналізом поведінки реципієнтів могли відразу обрати алгоритми, які їм потрібні.

Практичне значення одержаних результатів

За результатами проведеного аналізу ефективності реалізації алгоритмів Data mining було сформовано рекомендації щодо їх використання малими компаніями.

Ключові слова

MS SQL Server, Microsoft Visual Studio, Data mining.

РЕФЕРАТ
НА МАГИСТЕРСКУЮ ДИССЕРТАЦИЮ

выполненную на тему: Анализ эффективности реализации алгоритмов Data mining с использованием сервиса Microsoft SQL Server
студентом: Ниценко Андреем Сергеевичем

Работа выполнена на 98 страницах, содержит 61 иллюстраций, 22 таблицы. При подготовке использовалась информация из 33 источников.

Актуальность темы

На сегодняшний день данных, которые нужно анализировать становится все больше и больше. Компании имеют большие объемы информации, но иметь информацию мало, нужно еще и знать какими именно способами ее обрабатывать и анализировать.

Именно поэтому данное исследование имеет большие перспективы, поскольку понимание каждого алгоритма Data mining и области его лучшего использования будет или упрощать анализ для таких компаний, или давать большую точность полученных результатов.

Цель и задачи исследования

Целью данной работы является реализация и исследование эффективности реализации алгоритмов Data mining, которые предоставляет сервис MS SQL Server и Microsoft Visual Studio, чтобы понять для каких конкретных задач и с каким объемом данных будет лучшим тот или иной алгоритм.

Решение поставленных задач и достигнутых результатах

В работе рассмотрены алгоритмы (алгоритм кластеризации, алгоритм дерева принятия решений, алгоритм линейной регрессии, алгоритм взаимосвязей, упрощенный алгоритм Байеса, алгоритм нейронной сети, алгоритм кластеризации последовательностей, алгоритм временных рядов), которые предоставляет сервис MS SQL Server и Microsoft Visual Studio, реализовано каждый из них.

Избран задачи (прогнозирование дискретного атрибута, прогнозирования непрерывного атрибута, прогнозирования последовательности, нахождение групп общих элементов в транзакциях), которые нужно реализовать и избран алгоритмы в соответствии со спецификой задач. Реализованы алгоритмы Data mining. В ходе анализа оценки эффективности реализации алгоритмов были сформированы рекомендации по их использованию для различных задач.**Объект исследований**

Алгоритмы Data mining в среде MS SQL Server и Microsoft Visual Studio.

Предмет исследований

Эффективность реализации алгоритмов Data mining в среде MS SQL Server и Microsoft Visual Studio.

Методы исследований

Для решения указанной проблемы в работе проведено сравнение реализованных алгоритмов Data mining, выделено задачи на основе которых будет проведен анализ и выделены основные функциональные особенности каждого метода.

Научная новизна

Научная новизна заключается в том, что после реализации каждого алгоритма для конкретной задачи установлено, какой именно алгоритм лучше подходит для каждой задачи, чтобы в будущем малые компании, которые будут заниматься анализом поведения реципиентов могли сразу выбрать алгоритмы, которые им нужны.

Практическое значение полученных результатов

По результатам проведенного анализа эффективности реализации алгоритмов Data mining были сформированы рекомендации по их использованию малыми компаниями.

Ключевые слова

MS SQL Server, Microsoft Visual Studio, Data mining.

ABSTRACT
ON MASTER'S THESIS

on topic: Virtual office of the teacher-researcher

student: Andrii S. Nytsenko

The work is done on 98 pages, contains 61 illustrations, 22 tables. Information from 33 sources was used in the preparation.

Topicality

Today, the data that needs to be analyzed is becoming more and more. Companies have large amounts of information, but it is not enough to have information, you also need to know exactly how to process and analyze it.

That is why this study has great prospects, because understanding each Data mining algorithm and the areas of its best use will either simplify the analysis for such companies, or give greater accuracy.

Purpose

The purpose of this work is to implement and study the effectiveness of the implementation of Data mining algorithms provided by MS SQL Server and Microsoft Visual Studio, to understand for which specific tasks and with what amount of data will be the best algorithm.

Solution

The algorithms (clustering algorithm, decision tree algorithm, linear regression algorithm, interconnection algorithm, simplified Bayesian algorithm, neural network algorithm, sequence clustering algorithm, time series algorithm) that MS S Server provides to the service each of them is implemented.

Tasks (prediction of a discrete attribute, prediction of a continuous attribute, prediction of a sequence, finding groups of common elements in transactions) are selected, which need to be implemented and algorithms are selected according to the specifics of the tasks. Data mining algorithms are implemented. In the course of the

analysis of the evaluation of the efficiency of the algorithms implementation, recommendations on their use for various tasks were formed.

The object of research

Data mining algorithms in MS SQL Server and Microsoft Visual Studio.

The subject of research

Efficiency of implementation of Data mining algorithms in the environment of MS SQL Server and Microsoft Visual Studio.

Research methods

To solve this problem, the paper compares the implemented data mining algorithms, highlights the tasks on the basis of which the analysis will be performed and highlights the main functional features of each algorithm.

Scientific novelty

The scientific novelty is that after the implementation of each algorithm for a specific task, it is determined which algorithm is best suited for each task, so that in the future small companies that will analyze the behavior of recipients can immediately choose the algorithms they need.

The practical significance of the results obtained

Based on the results of the analysis of the effectiveness of the implementation of Data mining algorithms, recommendations for their use by small companies were formed.

Keywords

MS SQL Server, Microsoft Visual Studio, Data mining.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних.

ПЗ – програмний засіб.

СУБД – система управління базами даних.

SSAS – SQL Server Analysis Services (інструмент аналітичної обробки та аналізу даних).

BI – Business Intelligence (інструмент, який може допомогти у виявленні проблем та можливостей для зростання та розвитку)

ВСТУП

У сучасному світі з кожним роком обсяг інформації збільшується. На сьогоднішній день існує багато теорій, коли відбудеться інформаційний вибух. Динаміка останніх років показує циклічний зростання кількості інформації і збільшення даних в два рази протягом кожних двох років. У поточних реалії компанії середніх розмірів починають стикатися з проблемою обробки великого потоку даних, яка до того ж є досить розрізненою і неструктурованою інформацією. Впроваджують системи зберігання даних, озеро даних, компанії навчаються зберігати дані, але не отримувати з них вигоду. У цій роботі будуть дані відповіді на наступні актуальні питання:

- Які завдання бізнесу здатні вирішувати алгоритми Data mining?
- Що таке алгоритми Data mining?
- Навіщо потріні рішення класу BI?

1 ОГЛЯД ТЕХНОЛОГІЙ ТА АЛГОРИТМІВ DATA MINING З ВИКОРИСТАННЯМ MS SQL SERVER

1.1 Служба SSAS та її інтелектуальний аналіз даних

SQL Server надавав можливість інтелектуального аналізу в Analysis Services та був лідером у прогнозній аналітиці 2000. Формування інтегрованої платформи для прогнозної аналітики відбувається за допомогою сполучення Reporting Services, SQL Server Data та Integration Services. SQL Server Data mining поєднує стандартні алгоритми, нейронні мережі, K-середні і моделі кластеризації EM, лінійну і логістичну регресії та класифікатори алгоритму Байеса. Для спеціального уточнення, оцінки та безпосередньо полегшення розробки, кожна із моделей має візуальні елементи, що є інтегрованими [1]. Це допомагає у вирішенні проблем.

1.2 Інтелектуальний аналіз даних і його переваги

Машинне навчання та прогнозна аналітика – саме так називають ще інтелектуальний аналіз. Для виокремлення закономірностей у даних у ньому застосовуються статистичні принципи, що вже є добре дослідженими. Використавши алгоритми Analysis Services для своїх даних, можна виокремлювати закономірності, аналізувати події з їх послідовністю у наборах даних, а також прогнозувати тенденції.

Для аналізу і складання звітів переважно використовуються засоби SQL Server 2017, адже значна кількість користувачів стверджують, що вони доступні, мають між собою інтеграцію та широкі можливості [2].

1.3 Інтелектуальний аналіз даних та його ключові функції

SQL Server Data mining для підтримки вбудованих рішень включає такі функції [3]:

- Декілька алгоритмів, що можна налаштувати: SQL Server підтримує розробку власних користувальницьких алгоритмів користування, що налаштовуються.
- Декілька джерел даних: для інтелектуального аналізу можливе застосування різних джерел даних(текстові файли, таблиці), до того ж є доступним і аналіз кубів OLAP, що були зроблені в Analysis Services.
- Деталізація та запити: для інтеграції запитів, прогнозованих у додатки, за допомогою аналізу даних SQL Server надається мова для DMX. Можливе виконання деталізації варіантів після отримання шаблонів із моделей та статистичних даних.
- Інфраструктура перевірки моделей: для варто просто скористатися статистичними засобами такими, як точкові діаграми, матриці класифікації та перехресна перевірка. Функціонал створення та управління навчальними та наборами для перевірки дуже простий.
- Управління, звітність та очистка даних: для цього Integration Services пропонує потужні засоби для очищення та профілювання даних. Для цього варто створювати процеси ETL, а ssISnoversion полегшує оновлення та навчання моделей.
- Клієнтські кошти: додатково до середовищ проектування та розробки , що включені в SQL Server були додані функції перегляду й створення моделей та запити для їх виконання (робота з Excel)
- API-інтерфейси та підтримка мови сценаріїв: об'єкти інтелектуального аналізу даних програмовані. Розширення PowerShell для Analysis Services або ж сценарії через XMLA та MDX. Швидке створення запитів і сценаріїв відбувається за допомогою DMX.

- Розгортання та захист: захист відбувається на базі ролей через Analysis Services. Характеризується простим інтерфейсом розгортання та взаємодії моделей, що дозволяє користувачам створювати прогнозування.

1.4 Інтелектуальний аналіз даних та побудова моделі

Інтелектуальний аналіз даних – процес, що допомагає виявити відомості в великих наборах даних, що є приданими. В Data mining застосовується математичний аналіз для того, щоб виявити тенденції та закономірності, що існують в даних, варто використати математичний аналіз. Оскільки буває надмірний обсяг чи надто складні зв'язки, то ці закономірності неможливо виявити за допомогою традиційного перегляду [4].

Ці тренди та закономірності і складають модель інтелектуального аналізу даних. Такі моделі застосовуються у конкретних сценаріях [5]. Наприклад:

- Пошук послідовностей: прогноз події та аналіз вибору замовників, коли робляться покупки
- Прогнозування: оцінка продажів, прогнозування часу простою сервера
- Рекомендації: створення рекомендацій та ідентифікація продуктів, що можуть бути продані оптом
- Імовірність та ризик: визначення точки рівноваги , відсортування замовників, які є придатними для цільової розсилки
- Групування: прогнозування спільних рис, аналіз та розмежування подій чи замовників на певні кластери

Побудова моделі даного аналізу даних – частина процесу, що складає такі завдання: формулює питання і створює модель для відповіді на них [6]. Покрокова інструкція заключається в тому, щоб:

1. Установити задачі;
2. Підготувати дані;
3. Вивчити їх;

4. Створити моделі;
5. Проаналізувати та перевірити їх;
6. Розгорнути та оновити моделі.

Далі можна спостерігати опис зв'язків між етапами процесу і технологіями Microsoft SQL Server (рис. 1.1).

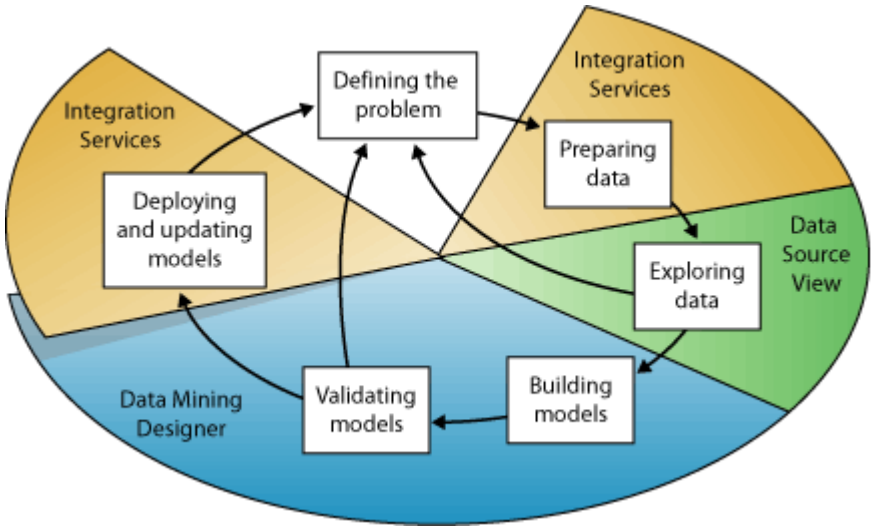


Рисунок 1.1 – Зв’язки між етапами процесу і технологіями MS SQL Server

Наступний процес циклічний, адже аналітична модель даних створюється динамічно та з повтореннями. Іноді доводиться робити пошук додаткових даних, адже користувач при перегляді, може помітити, що даних замало для створення потрібних моделей інтелектуального аналізу даних. Також іноді виникає нюанс у відсутності адекватної потрібної відповіді на питання, після створення кількох моделей, аби це виправити постановка завдання трактується по-іншому. Також час від часу буває необхідно оновлювати вже відкриті моделі, бо надходять нові дані. Як згадувалося на початку про циклічність, щоб утворити оптимальну модель, варто кілька раз повторити операції [7].

Інтелектуальний аналіз даних Microsoft SQL Server допомагає створити за допомогою інтегрованого середовища моделей інтелектуального аналізу даних. Програма SQL Server Development Studio входить до цього середовища. Вона

включає в себе засоби створення запитів та алгоритми власне інтелектуального аналізу даних, що значно спрощує прийняття рішення для певних проектів. Також тут є SQL Server Management Studio – компонент, що має засоби для управління об’єктами та пошуку моделей інтелектуального аналізу даних [8].

1.4.1 Визначення проблеми

Перший крок – чітке визначення проблеми. Далі варто розглянути різноманітні методи для вирішення проблеми із використанням даних (рис. 1.2).

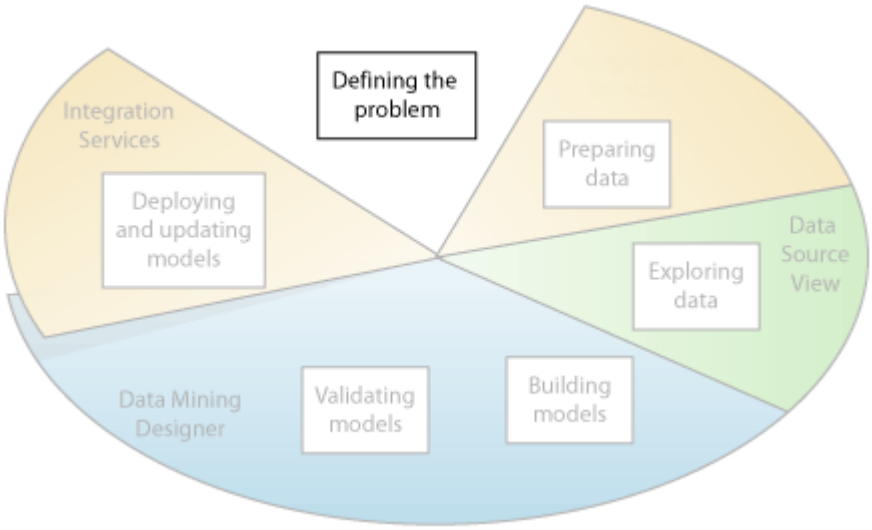


Рисунок 1.2 – Етап визначення проблеми

Цей крок визначає завдання для проекту, включає в себе визначення області проблеми та метрики (згідно з ними і відбувається оцінювання моделі), а також необхідним є аналіз бізнес-вимог. Ці завдання складають питання, подані нижче [9].

- Що та які типи зв'язків треба відшукати?
- Яким повинен бути спрогнозований результат?

- Чи можна знайти взаємозв’язок та змістовні закономірності без потреби прогнозування за допомогою моделі інтелектуального аналізу даних?
- Як відбувається розподіл даних, чи є вони сезонними?
- Чи відображають дані бізнес-процеси?
- Які види даних повинні бути та які відомості мають розташовуватися у полях таблиці? Чи є взаємозв’язок між ними?

Спочатку варто дослідити потреби користувачів та дослідити рівень доступності даних, аби дати точні відповіді. При невідповідності даних вимогам та потребам користувачів змінюється визначення проекту. Додатково розглядаються способи фіксації результатів моделі в основних показниках ефективності, вони застосовуються для оцінювання діяльності бізнесу [10].

1.4.2 Підготовка даних

Другий кроком – поєднання та очищення даних, що визначені попереднім кроком (рис. 1.3).

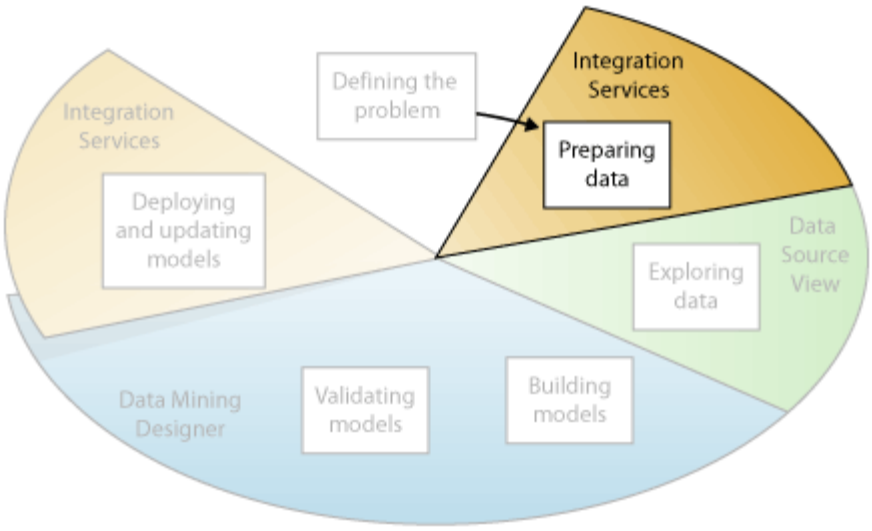


Рисунок 1.3 – Етап підготовки даних

Дані можуть не мати записів або містити їх із помилками. Також вони можуть зберігатися у різних форматах та частинах компанії. Згідно з даними, клієнт може робити покупки в магазині іншого міста чи то країни або ж взагалі придбати продукт до реалізації його на ринку.

Очищення – це пошук у даних прихованих залежностей, підбір стовпців та виокремлення найточніших джерел для аналізу, а не як всі звикли думати, що це просто відсіювання невірних даних. Наприклад, чи варто знати дату замовлення чи/або відвантаження? Що найбільше впливає на продаж : знижка на товар, його кількість чи кінцева ціна? Отже, на результати моделі впливають нібито незалежні вхідні параметри, що становлять хороший взаємозв'язок, а також помилкові неповні дані [11].

У першу чергу для побудови моделей інтелектуального аналізу даних варто виявити подібні проблеми, а потім зрозуміти методи їх усунення. Так як під час аналізу даних доводиться працювати зі значними наборами даних та немає можливості відслідковувати кожну із транзакцій на предмет якості даних, треба користатися засобами фільтрації та автоматичного очищення даних. Такі є в Integration Services, Master Data Services, Microsoft SQL Server 2012 чи SQL Server Data Quality Services.

Варто знати, що не обов'язково зберігати дані в реляційній базі чи в кубі аналітичної обробки в мережі (OLAP), проте їх використовують у якості джерел даних. Для інтелектуальний аналіз даних застосовується будь-яке джерело із категорії Analysis Services. Це можуть бути дані із різних зовнішніх постачальників, книги Excel та будь-які текстові файли [12].

1.4.3 Перегляд даних

Третій крок процесу інтелектуального аналізу даних полягає у перегляді (рис. 1.4).

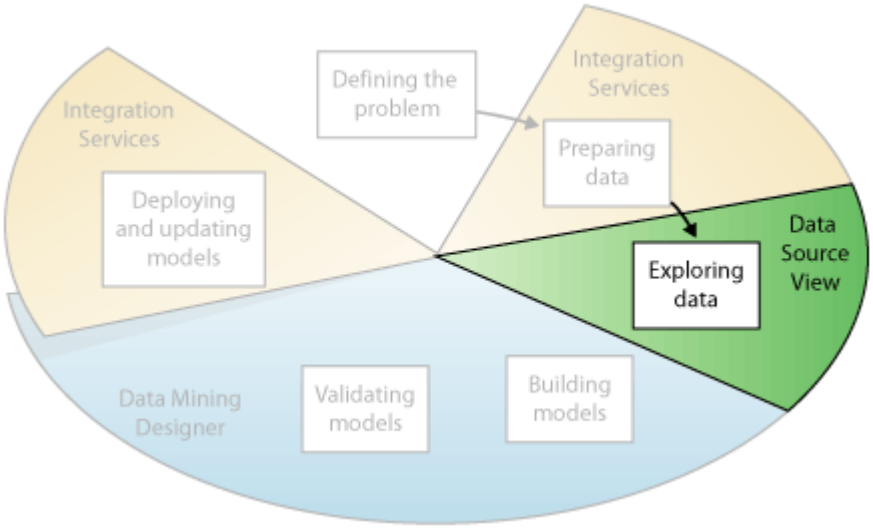


Рисунок 1.4 – Етап перегляду даних

Варто розуміти дані, аби приймати правильно рішення при створенні моделей інтелектуального аналізу даних. Для дослідження необхідно враховувати розподіл даних, розраховувати максимальне і мінімальне значення. По них можна визначити чи є репрезентативною вибірка даних для наявних клієнтів або бізнес-процесів. Для цього також часто змінюють припущення, що закладені в основі очікуваних результатів. Стандартне відхилення та інші характеристики розподілу чи ймовірне стандартне відхилення говорять про точність та стабільність. Якщо стандарт відхилення завеликий, то це свідчить, що нові дані можуть сприяти вдосконаленню моделі. Вони можуть створювати точний чи спотворений образ наявної проблеми, що ускладнює сам процес підбору необхідної моделі [13].

Якщо аналізувати дані з власної точки зору щодо бізнес-проблеми, можна виявляти нові помилки в їх наборі, а це відкриває можливості до створення стратегії для скасування проблем або ж допомагає просто пізнати краще саму модель поведінки, яка характерна для бізнесу.

1.4.4 Створення моделей

Четвертий кроком аналізу даних – побудова моделей. Застосувавши всю інформацію, що розглядалася у попередньому кроці, можна створити модель (рис. 1.5).

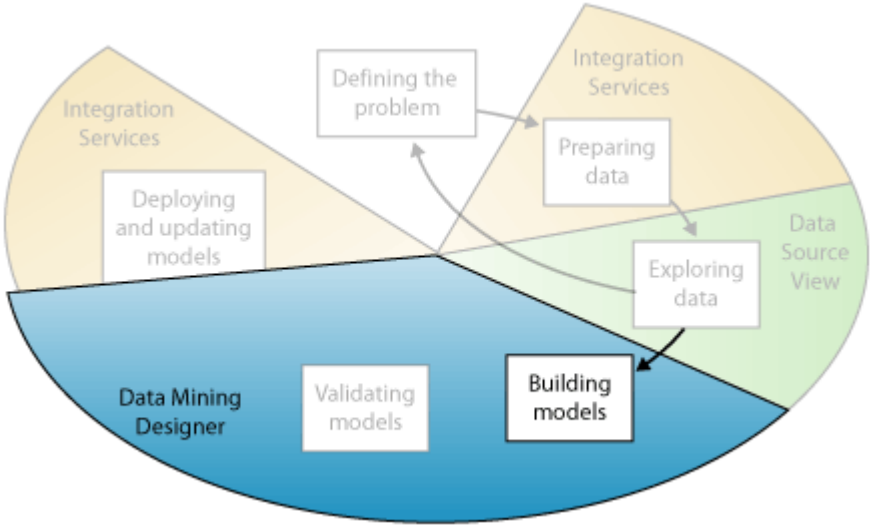


Рисунок 1.5 – Етап створення моделей

Спочатку відбираються стовпці даних, застосування яких буде необхідним у створенні структури інтелектуального аналізу даних. Вона не містить жодних, хоч і пов’язана з джерелом. При обробці структури інтелектуального аналізу Analysis Services застосовуються статистичні дані та вирази. Вони, у свою чергу, є спільними для будь-якої моделі, пов’язаної з цією структурою [14].

Часто обробку моделі іменують навчанням, а саму аналіз – це певний контейнер, що програмує стовпці для вхідних даних. Обробка моделі – це процес, під час якого використовується математичний алгоритм до даних у структурі для виявлення закономірностей. Саме вибір навчальних даних, конкретний алгоритм та його конфігурацію взаємозалежні із закономірностями. У SQL Server 2017 є різноманітні алгоритми, де кожен задається для різного тип завдань і, власне, створює особливу модель.

Для налаштування використовуються фільтри для даних, аби була задіяна виключно їх підмножина, а також деякі параметри. Коли дані проходять через модель, то об’єкт матиме певні зведені дані і закономірності, їх можна використовувати для прогнозування.

SQL Server Data Tools допомагає вирахувати нову модель , також можна скористатися мовою розширень [15].

При цьому не варто забувати, коли змінюються дані, слід оновлювати модель і структуру інтелектуального аналізу даних.

Коли це відбувається повторно та динамічно, тоді Analysis Services отримують дані з джерела. При цьому можна лишати моделі незмінними або ж оновлювати. Останній процес означатиме циклічне навчання уже за допомогою нових даних [16].

1.4.5 Дослідження моделей

П'ятий крок – дослідження побудованих моделей інтелектуального аналізу даних та тестування їх ефективності (рис. 1.6).

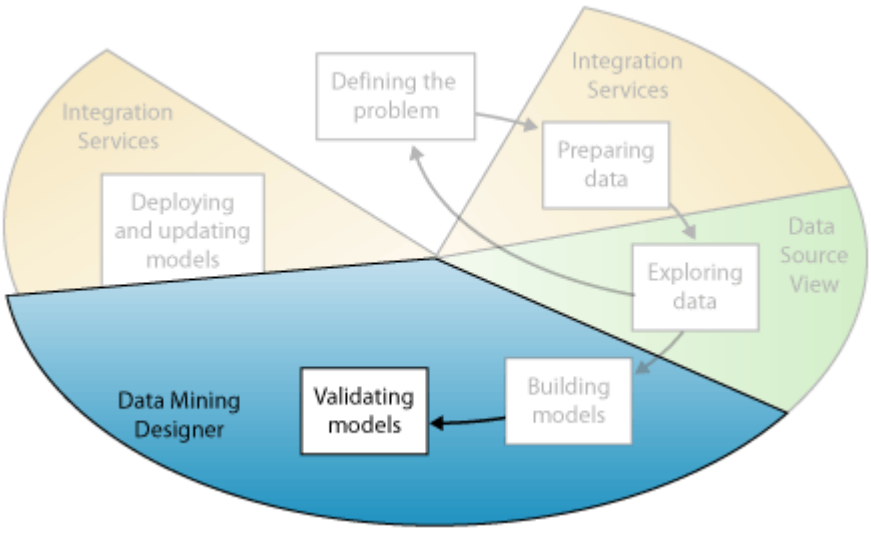


Рисунок 1.6 – Етап дослідження моделей

Перед тим, як розгорнути модель у робочому середовищі, треба обов'язково протестувати ефективність її роботи. Крім цього, щоб забезпечити найкращі результати, створюються одразу кілька моделей з неідентичним конфігураціями, після цього і проходить тестування.

Analysis Services надають засоби, що полегшують розмежування даних для перевірки та навчання. Це дає можливість краще оцінити продуктивність моделей. Набір даних для навчання використовується У ході побудови застосовується набір для навчання, а наступні – відповідно, для перевірки. Це секціонування виконується автоматично під час створення моделі інтелектуального аналізу даних [17].

Засоби перегляду в середовищі SQL Server Data Tools дозволяють дослідити закономірності та тенденції. Точність прогнозів, що створюються за допомогою моделей, перевіряються засобами конструктора. До таких належать матриця класифікації та діаграма точності прогнозів. Статистичний метод – перехресна перевіркою для створення підмножини даних і перевірки модель по кожній із підмножин. Він дозволяє перевірити чи придатність моделі обмежена на наявні дані.

Іноді доводить повертатися до попереднього кроку, якщо жодна з моделей, не має потрібної ефективності, також можна переформулювати завдання.

1.4.6 Розгортання й оновлення моделей

Заключний крок – це розгортання найефективніших моделей в робочому середовищі (рис. 1.7).

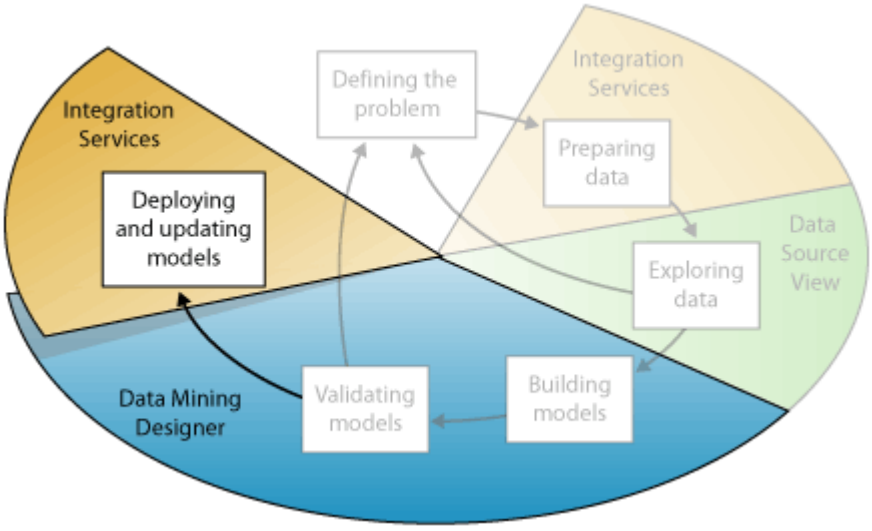


Рисунок 1.7 – Етап розгортання і оновлення моделей

Після цього процесу користувачу відкриваються можливості до виконання безлічі завдань таких як:

- Створення запитів вмісту для отримання правил, статистики або формул.
- Застосування моделі для прогнозів, що потім використовуються у прийнятті бізнес-рішень. Для створення запитів використовуємо SQL Server.
- Впровадження у додаток функцій інтелектуального аналізу даних. Можна використовувати об'єкти AMO, що включають набір об'єктів. Вони застосовуються для зміни, створення, обробки і видалення структур та моделей аналізу. Також можна надсилати повідомлення XML для аналітики (XMLA) в Analysis Services.
- Integration Services застосовуються для створення пакету. У цьому випадку модель аналізу розподіляє вхідні дані у різні таблиці. Наприклад, при постійному оновленні модель інтелектуального аналізу даних може паралельно використовуватися з Integration Services. Це допомагає розсортувати клієнтів, що куплять чи навпаки не придбають товар.

- Оновлення моделей після перегляду і аналізу. Після оновлення повторно проводиться обробка моделей.
- Створення звіту.
- До стратегії розгортання обов’язкового входить динамічне оновлення моделей.

1.5 Аналіз існуючих алгоритмів інтелектуального аналізу даних в MS SQL Server

В інтелектуальному аналізі даних (Analysis Services) алгоритм означає набір обчислень і евристики. Він і створює модель на основі даних. Для цього спочатку триває аналіз даних за допомогою пошуку певних тенденцій та закономірностей. Щоб підібрати оптимальні параметри для створення моделі, алгоритм використовує результати аналізу до безлічі ітерацій. Наступним кроком є застосування цих параметрів до усього набору, це дозволяє отримати докладну статистику та виявити закономірності, що є придатними для використання [18].

Модель, що створена алгоритмом з отриманих даних набуває таких форм:

- Математична модель, котра прогнозує продаж;
- Набір кластерів, що описують зв'язки варіантів;
- Дерево рішень, яке передбачає результат і оцінює вплив різних критеріїв на нього;
- Групування продуктів в транзакції та ймовірність одночасної покупки продуктів.

SQL Server використовує вивчені та найпопулярніші методи виявлення закономірностей в даних. Наприклад, алгоритм кластеризації методом К-середніх – один із найстаріших алгоритмів, його застосовують у багатьох інструментах та з різними параметрами. Даний алгоритм, що реалізований в

інтелектуальному аналізі даних SQL Server, був розроблений групою Microsoft Research, а потім його оптимізували для роботи з Analysis Services. Усі алгоритми Майкрософт доступні для налаштування та програмування з використанням наданих API. Також можна також автоматизувати навчання та створення моделей за допомогою компонентів інтелектуального аналізу даних у службах Integration Services [19].

До того ж підтримується і використання OLE DB. Можна також розробляти власні алгоритми, а, зареєструвавши їх в якості служб, потім використовувати на платформі SQL Server.

У певних аналітичних задачах іноді складно обрати правильний алгоритм. Деякі з них можуть видавати більше одного результату. Прикладом є використання алгоритму дерева прийняття рішень не лише для прогнозування, але й для зменшення кількості стовпців в наборі даних, тому що воно може розпізнавати стовпці, що не мають впливу на кінцеву модель інтелектуального аналізу даних.

SQL Server Data mining має такі типи алгоритмів:

- Регресивні – прогнозують одну або декілька безперервних числових змінних(прибутку чи збитків);
- Алгоритми класифікації прогнозують одну або кілька дискретних змінних при інших атрибутах;
- Алгоритми сегментації сортують дані на кластери або групи;
- Алгоритми взаємозв'язків сприяють пошуку кореляції між різними атрибутами в наборі даних. Найчастіше застосовується правило взаємозв'язку, що можуть використовуватися для аналізу кошика споживача;
- Алгоритми аналізу послідовностей здатні узагальнювати послідовності, котрі трапляються часто в даних (як серія подій або переходів по веб-сайту, що були зареєстровані).

Проте попри все користувач може використовувати багато алгоритмів. Аналітики з досвідом можуть працювати за відточеною схемою одного алгоритму для того, щоб виявити найефективніші змінні дані, а вже після – обирають інший алгоритм, щоб провести прогнозування результату на їх основі. SQL Server Data mining дає можливість будувати кілька моделей на основі однієї структури інтелектуального аналізу таким чином, щоб у межах одного рішення аналізу даних застосовувався алгоритм кластеризації, модель дерева рішень, а також – алгоритму Байеса для отримання різних даних. Також кілька алгоритмів для окремих завдань можуть використовуватися у одному рішенні. Наприклад, алгоритм нейронної мережі допомагає провести аналіз чинників, що впливають на прогнози, а фінансові прогнози отримуються за допомогою регресії [20].

1.5.1 Алгоритм кластеризації

Це алгоритм сегментації, що виконує ітерацію варіантів у наборі даних, для згрупування їх у кластери з подібними характеристиками. Групування корисно застосовувати для перегляду даних, а також, щоб виявити у них аномалії та створити прогнози.

Моделі кластеризації визначають зв'язок у наборі даних, що неможливо отримати лише за допомогою випадкового спостереження. Наприклад, за логічним ланцюгом можна здогадатися, що люди, котрі їдуть велосипедом на роботу, не обов'язково повинні жити далеко. У даному випадку алгоритм знаходить інші характеристики велосипедистів. На наступній діаграмі (рис. 1.8) кластер А відповідає людям, що добираються на машині до роботи, кластер Б відповідає людям, що добираються на велосипеді на роботу [21].

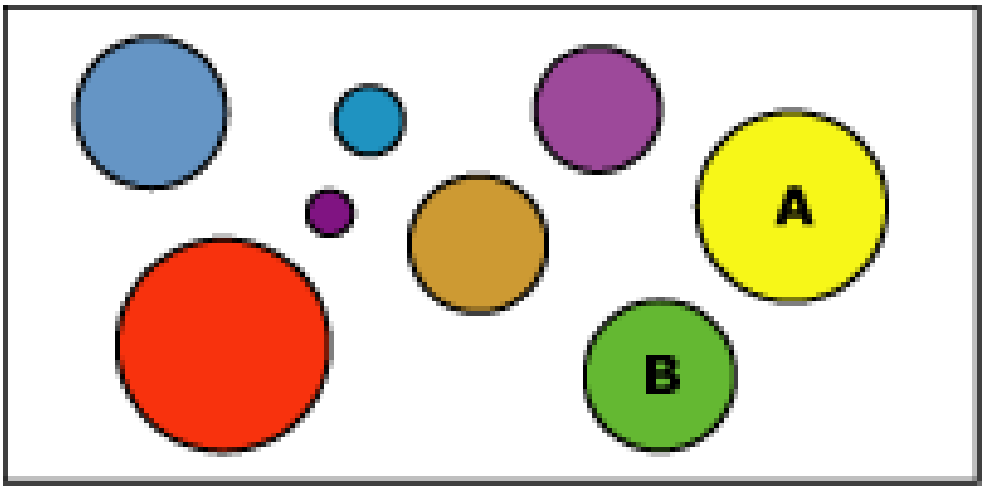


Рисунок 1.8 – Приклад алгоритму кластеризації

Алгоритм кластеризації є відмінним від інших алгоритмів інтелектуального аналізу даних. У ньому треба обов’язково призначати стовпець прогнозування, що не є необхідним, наприклад, у алгоритмі дерева прийняття рішень. Алгоритм кластеризації навчає модель на основі кластерів, що ідентифікуються алгоритмом та на основі зв'язків, які існують у даних [22].

Принцип роботи алгоритму: у наборі даних спочатку триває визначення зв’язків за алгоритмом кластеризації, а потім триває формування ряду кластерів на основі цих зв'язків. Як відбувається групування даних можна спостерігати у точковій діаграмі. Вона зображає всі можливі варіанти в наборі даних, і кожен із них є певною точкою. А показ зв’язків відбувається за допомогою групи кластерів.

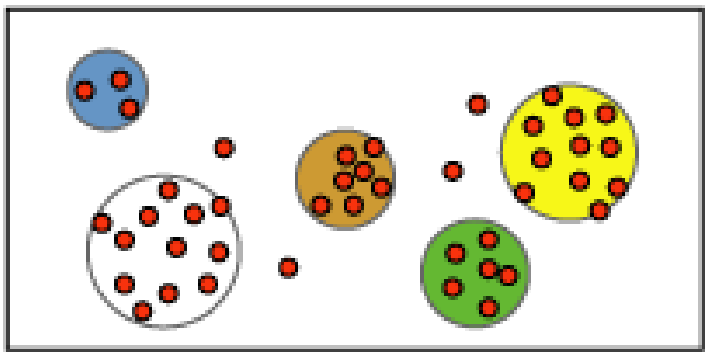


Рисунок 1.9 – Принцип роботи алгоритму кластеризації

Спочатку після визначення кластерів алгоритм обчислює, як кластери являють групування точок. Наступним кроком є повторне визначення групування, яке створює кластери для кращої уяви даних. Алгоритм циклічно доки поріг покращення кластерів дійде до максимуму.

Можна налаштовувати роботу даного алгоритму, вибираючи конкретний метод об'єднання в кластери, обмежуючи максимальну кількість кластерів або змінюючи розмір множини, необхідну для створення кластера. Є два популярних метода кластеризації, що стосуються цього алгоритму – метод максимізації очікувань та кластеризація методом К-середніх [23].

При підготовці цих даних варто не забувати про вимоги до конкретного алгоритму:

- Вхідні стовпці – у кожній моделі повинен бути хоча б один, у якому є значення, що застосовують для формування кластерів. Кількість стовпців обтяжує процес та час навчання моделі.
- Одиничний ключовий стовпець – текстовий або числовий стовпець, що визначає кожен запис. Не можна використовувати складові ключі.
- Необов'язковий стовпець – передбачена можливість додавання прогнозованого стовпця з даними різних типів. Якщо треба передбачити дохід замовника шляхом кластеризації за віком та регіоном, то задається дохід такий як PredictOnly і варто ввести інші стовпці, де буде вказано додатково про регіон.

1.5.2 Алгоритм дерева прийняття рішень

Це алгоритм класифікації та регресії для використання в прогнозному моделюванні безперервних та дискретних атрибутів.

Для них алгоритм здійснює прогнозування, що базується на зв'язку між вхідними стовпцями в наборі даних. Алгоритм застосовує «стан» - значення цих стовпців для прогнозування станів стовпчика. Алгоритм розпізнає вхідні

стовпці, які корельовані зі стовпцем прогнозування. Наприклад, якщо покупці велосипеда 90 % молода аудиторія, то вік грає хороше значення у прогнозуванні. Дерево рішень створює прогнозування на основі цієї тенденції задля конкретного результату. Лінійна регресія використовується для встановлення місця розбивки дерева прийняття рішень.

Якщо декілька полів вказані як прогнозовані або якщо у вхідних даних міститься прогнозована таблиця або кілька стовпців, тоді для кожного прогнозованого стовпця алгоритм будує окреме дерево рішень.

Принцип роботи алгоритму: шляхом створення ряду розбиття в дереві будується модель інтелектуального аналізу даних. Вони презентуються як вузли. До моделі щоразу додається вузол за допомогою алгоритму. Це відбувається тоді, коли стає ясно, що у вхідному стовпці прослідковується суттєва кореляція з прогнозованим стовпцем. Способи для визначення розбиття відрізняються, а відмінністю є прогнозування дискретного чи безперервного стовпців.

Для вибору найкорисніших атрибутів алгоритмом дерева прийняття рішень використовується функція, що відсортовує компоненти. Їх вибір використовується всіма алгоритмами служб SQL Server. Це значно покращує продуктивність та якість аналізу. Щоб не використовувати процесорний час дарма і користуються вибором компонентів. Обробка займає багато часу або ж призводить до засмічення пам’яті, якщо у модель додається забагато прогнозованих або вхідних атрибутів. Для методів визначення необхідності розбивки дерева відносять стандартні метрики для Байесових та ентропії.

Надмірна чутливість до невеликих розбіжностей у даних і є найбільшою проблемою в моделях. Тоді її називають надто навченою або переоснащеною. Така модель зводиться до інших наборів даних. Аби запобігти появі невірних у наборах даних зв’язків застосовують методики, що контролюють ріст дерева.

Алгоритм дерева прийняття рішень можна використовувати для вирішення наступних задач:

1) Прогнозування дискретних стовпців

Гістограми демонструють способи, за допомогою яких і вибудовується алгоритм дерева для дискретного прогнозованого стовпця. На наступній діаграмі (рис. 1.10) побудовано стовпець «Покупці велосипедів» з вхідним стовпцем «Вік». Гістограма «Б» свідчить про наміри людини за віком чи придбає вона дану річ.

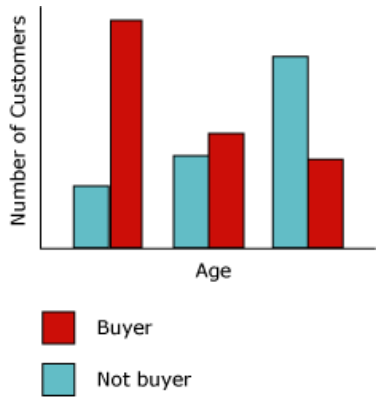


Рисунок 1.10 – Кореляція між вірогідністю покупки велосипеду і віком покупця

Продемонстрована кореляція призведе до створення нового вузла моделі (рис.1.11).

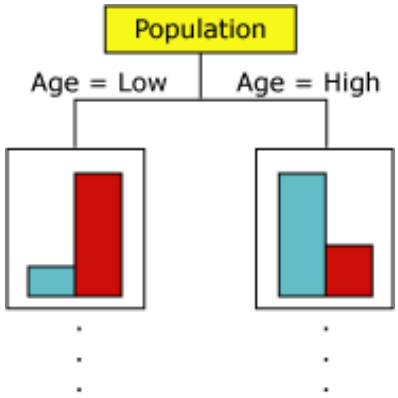


Рисунок 1.11 – Створення нового вузла моделі

Деревовидна структура створюється саме через додавання до моделі нових вузлів. При продовженні зростання моделі алгоритм розглядає Усі стовпці розглядаються при продовженні зростання певної моделі.

2) Прогнозування безперервних стовпців

У кожному вузлі міститься регресійна формула. При побудові дерева алгоритмом рішень Microsoft відбувається безперервне прогнозування стовпців (рис. 1.12).

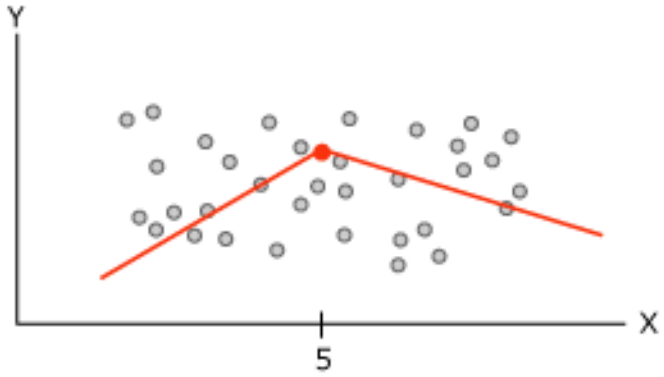


Рисунок 1.12 – Побудова дерева прийняття рішень

Зазвичай за стандартом у моделі регресії роблять спробу отримати формулу, котра презентує зв'язок даних та тенденції. Проте формула поодинці може неправильно показувати неоднорідність складних даних. Через це шукаються лінійні сегменти дерева та створюються окремі формули для них. Виходячи із цього, можна зрозуміти, що найточніше наближення забезпечується розбиттям даних на кілька сегментів [24].

Ця діаграма (рис. 1.13) представляє собою дерево для моделі, що демонстрована на точковій діаграмі вище. Дана модель представляє для прогнозування дві формули: одну для лівої галузі (y = 0.5x * 5) іншу – для правої (y = 0.25x + 8.75). Точка перетину їх – точка нелінійності, у ній і розіб'ється вузол в моделі дерева рішень.

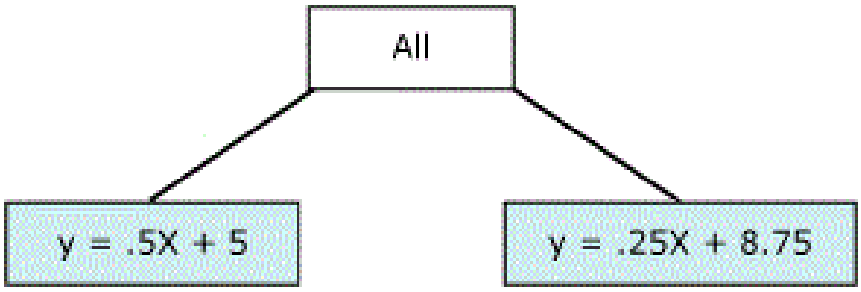


Рисунок 1.13 – Приклад моделі для дерева прийняття рішень

Це модель має лише 2 прості лінійні рівняння. Розбиття в дереві розташовується після вузла «All», проте не варто виключати варіант розбиття на будь-якому із рівнів. Поясненням є те, що формула може застосовуватися одразу для одного чи кількох вузлів. Наприклад, виходить формула для вузла – "вік і дохід клієнтів" та інша – "клієнти, яким потрібно долати значні відстані". Для перевірки формули сегмента чи конкретного вузла натискаємо на вузол.

Необхідно розуміти вимоги певних алгоритмів, готуючи дані для використання у моделі. Це можуть бути такі вимоги як метод їх використання або ж їх кількість.

Існують наступні вимоги:

- Вхідні стовпці можуть бути безперервними або дискретними. На час обробки безпосередньо мають вплив вхідні атрибути, а саме – їх кількість.
- Одиничний ключовий стовпець – текстовий або числовий стовпець, що визначає кожен запис. Важливо, що складові ключі не застосовуються.
- Прогнозований стовпець – їх у модель можна включати кілька різного типу. Варто пам’ятати, що час обробки збільшується разом із кількістю прогнозованих атрибутів.

1.5.3 Алгоритм лінійної регресії

Це різновид алгоритму дерева прийняття рішень, який допомагає визначити лінійний зв'язок між залежною і незалежною змінною та застосовувати його під час прогнозування.

Зв'язок має вигляд лінії із рядом даних. Найкраще лінійна подача даних спостерігається на наступній діаграмі (рис. 1.14).

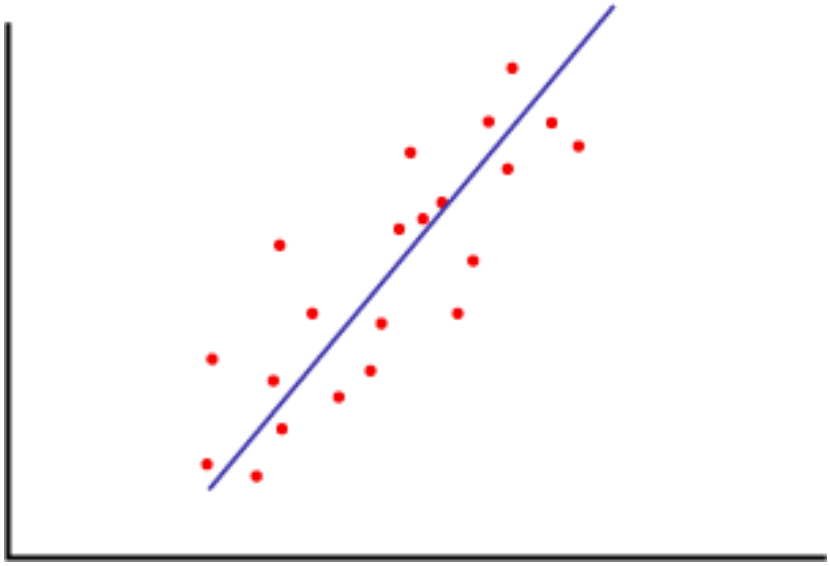


Рисунок 1.14 – Приклад моделі з застосуванням алгоритму лінійної регресії

Помилка, пов'язана з кожною точкою на діаграмі, відповідає відстані від лінії регресії; a і b – коефіцієнти в рівнянні регресії регулюють положення та кут лінії регресії. За допомогою підбору коефіцієнтів a і b можна отримувати регресивне рівняння, доки сума помилок не буде мінімальною.

У інших типах регресії застосовується кілька змінних та також нелінійні методи регресії. Лінійна регресія – загально відомий метод моделювання відповіді, він є корисним.

Лінійна регресія застосовується для визначення зв'язку між двома безперервними стовпцями. Наприклад, її можна використовувати для розрахування лінії тренду в даних продажів або виробничих даних. Для

розробки складніших моделей інтелектуального аналізу даних також використовується лінійна регресія. З нею можна провести оцінку зв'язків між стовпцями даних.

Попри існування багатьох різних способів для розрахунку все одно доцільно використовувати алгоритм лінійної регресії, тому обчислення і перевірка всіх можливих зв'язків між змінними проводиться автоматично. Також необов'язково задавати метод обчислення. Також значно спрощуються зв'язки в сценаріях, теж варто враховувати [25].

Алгоритм лінійної регресії – різновид алгоритму дерева прийняття рішень. При виборі його виникає особливий варіант алгоритму дерева прийняття рішень. Він містить параметри, котрі обмежують поведінку алгоритму і вимагають застосування певних типів даних на вході. При початковому проході у моделі застосовується весь набір даних, а от у стандартній моделі – дані розбиваються на менші підмножини або дерева кілька разів.

Необхідно враховувати вимоги конкретних алгоритмів при підготовці даних. Варто звертати увагу на обсяг даних та їх використання. Для моделей лінійної регресії застосовуються наступні вимоги:

- Одиничний ключовий стовпець – текстовий або числовий стовпець, що визначає кожен запис. Варто пам’ятати про недопущення складових ключів.
- Вхідні стовпці – мають містити безперервні числові дані та відповідний тип.
- Прогнозований стовпець – повинен бути хоча б один, але дозволено кілька, при умові наявності безперервних числових типів даних. Не можна застосовувати datetime.

Результати після обробки моделі зберігаються у вигляді набору статистичних даних разом із формулою лінійної регресії. Вона може бути застосована для розрахунку трендів.

1.5.4 Алгоритм взаємозв'язків

Алгоритм взаємозв'язків застосовується для механізмів вироблення рекомендацій. Механізм радить продукти користувачам на основі тих, до яких вони уже приходили та/або придбали. Даний алгоритм взаємозв'язків зручно застосовувати, щоб проаналізувати кошик споживача.

Моделі взаємозв'язків будуються на наборах даних, які включають ідентифікатори для варіантів і окремих їх елементів. Набір елементів – група елементів у варіанті. Ряди наборів елементів і правил складають модель взаємозв’язків. Правила, що визначені алгоритмом, можуть застосовуватися для прогнозування майбутніх покупок клієнтів на основі речей, що вже є в кошику. Далі демонструється ряд правил у наборі елементів (рис. 1.15).

Rule
Road Bottle Cage = Existing, Cycling Cap = Existing -> Water Bottle = Existing
Mountain-200 = Existing, Mountain Tire Tube = Existing -> HL Mountain Tire = Existing
Mountain-200 = Existing, Water Bottle = Existing -> Mountain Bottle Cage = Existing
Touring-1000 = Existing, Water Bottle = Existing -> Road Bottle Cage = Existing
Road-750 = Existing, Water Bottle = Existing -> Road Bottle Cage = Existing
Touring Tire = Existing, Sport-100 = Existing -> Touring Tire Tube = Existing

Рисунок 1.15 – Приклад правил з застосуванням алгоритму взаємозв’язків

Розглядаючи схему можна сказати, що алгоритм взаємозв'язків потенційно знаходить безліч правил всередині набору даних. Щоб описати набір елементів і правила, що він формує, алгоритм застосовує два параметри: ймовірність та потужність множини. Наприклад, якщо X і Y – це два елементи, що потенційно є у кошику для покупок, тоді параметр множини дорівнює

кількості варіантів у наборі даних з елементами X і Y. Застосовуючи параметр множини разом із MINIMUM_SUPPORT і MAXIMUM_SUPPORT, алгоритм керує кількістю наборів елементів, що створюються. Параметр ймовірності, котрий також називають достовірністю, надає частину варіантів у наборі даних із X і Y. Цей алгоритм керує кількістю сформованих правил, застосовуючи поєднання параметрів MINIMUM_PROBABILITY та ймовірності.

Алгоритм взаємозв'язків переглядає набір даних для пошуку елементів, що розташовуються в одному варіанті. Далі алгоритм групує в набори елементів знайдені та пов’язані між собою елементи. Це визначається за допомогою параметра MINIMUM_SUPPORT. Потім відбувається формування алгоритмом правил із наборів елементів. Вони застосовуються для прогнозування виявлення елемента в базі даних на основі наявності значущих елементів. Припустимо, у кошику клієнта є контейнер для фляги та туристичний набір, тоді ймовірність, що фляга теж буде куплена складає 81%.

При підготовці даних для використання в моделі правил взаємозв'язків, Потрібно враховувати вимоги до конкретного алгоритму при підготовці даних для використання в моделі правил взаємозв'язків. Також варто зазначити кількість та спосіб використання даних [26].

Вимоги до моделі правил взаємозв'язків такі:

- Вхідні стовпці – дискретні. Вони часто розміщуються в двох таблицях: в одній – це дані про клієнта, а в іншій таблиці – про його покупки. Ці дані вводять в модель за допомогою вкладеної таблиці.
- Одиничний ключовий стовпець – текстовий чи числовий, кожен визначає запис. Не можна допускати складові ключі.
- Єдиний прогнозований стовпець. Це може бути один ключовий стовпець вкладеної таблиці, наприклад, перелік придбаних продуктів. Значення мають бути дискретизованими або ж дискретними.

1.5.5 Спрощений алгоритм Байеса

Спрощений алгоритм Байеса – це алгоритм класифікації, заснований на основі теорем Байеса. Застосовується для прогнозованого та довільного моделювання. Алгоритм «спрощений», тому що застосовує методи Байеса без урахувань можливих залежностей.

Алгоритм не вимагає додаткових обчислень і застосовується для швидкого формування моделей інтелектуального аналізу даних для виявлення відносин між вхідними і прогнозованими стовпцями. Можна застосовувати для початкового дослідження даних, а потім використати результати для створення додаткових моделей інтелектуального аналізу з іншими алгоритмами, що є точними та мають у функціоналі вимоги до обчислень.

Приклад: відділ маркетингу вирішив розіслати листівки потенційним клієнтам. Для зниження собівартості прийняли рішення охопити тільки тих клієнтів, що дадуть відповідь, так як компанії відомі демографічні дані клієнтів та їх реакції на розсилки. Тож варто просто використати особисті дані про клієнтів із прогнозуванням можливості відповіді. Далі у рамках рекламної кампанії просто проводиться порівняння та аналіз «відомих» клієнтів і тих, що мають схожі ознаки і ті, що проявляли активність у минулому. Простіше кажучи, вимальовуємо картинку відмінностей між тими, хто уже придбав велосипед і тими, хто цього (ще) не зробив [27].

Відділ маркетингу може з легкістю спрогнозувати результат у рамках дослідження, скориставшись спрощеним алгоритмом Байеса. Застосовуючи метод перегляду спрощеного алгоритму Байеса в середовищі SQL Server Data Tools, відділ маркетингу може дослідити, які з вхідних стовпців мають хороший вплив на реакції.

Даний алгоритм розраховує ймовірність стану кожного вхідного стовпчика при такому ж стані прогнозованого стовпця.

Microsoft допоможе розібратися у системі цього методу з середовищем SQL Server Data Tools (рис. 1.16). Воно дає змогу в наочному режимі досліджувати, як розподіляються стани в межах цього алгоритму.

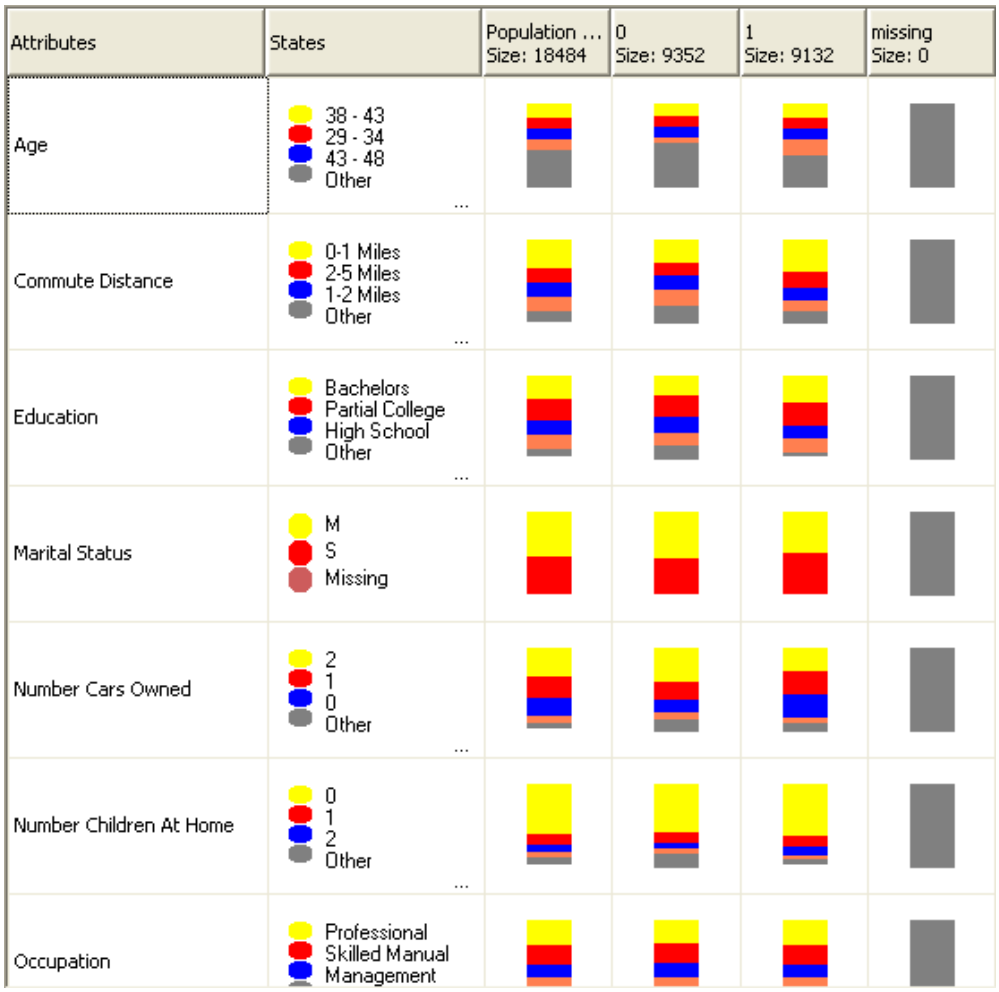


Рисунок 1.16 – Приклад розподілу станів в межах спрощеного алгоритму Байеса

Тут міститься список вхідних стовпців у наборі даних та показується розподіл стану кожного з стовпців при такому ж із станів прогнозованого стовпця.

Із цим поданням моделі визначаються також вхідні стовпці, що важливі для розмежування станів прогнозованого стовпця.

Наприклад, у рядку для поля Commute Distance розсортування вхідних значень наочно відрізняється для покупців і тих, хто не купує. Вхідне значення Commute Distance = 0-1 миля потенційно має вплив на результат прогнозу, це демонструють показані дані [28].

Також засіб перегляду відображає значення для окремих класів продуктів так, щоб можна було зрозуміти, що клієнти, які їдуть від дому до роботи від 1 до 2 миль, ймовірно придбають велосипед у 0,387, а не придбають – 0,287. Тут використано за допомогою алгоритму числові дані, які були отримані з ознак клієнтів, наприклад, дорога між домом і роботою.

Варто враховувати вимоги алгоритм при підготовці даних(а також обсяг і спосіб їх використання), що призначені для застосування в навчанні моделі спрощеного алгоритму Байеса.

Вимоги для моделі спрощеного алгоритму Байеса:

- Одиничний ключовий стовпець – текстовий чи числовий, що унікально визначає кожен запис. Не можна використовувати складові ключі.
- Вхідні стовпці повинні мати сегментовані значення або бути дискретними.
- Незалежні змінні – це обов’язковий фактор у алгоритмі Байеса, тут варто забезпечити незалежність вхідних атрибутів один від одного. Це необхідно застосувати при прогнозуванні. При взаємопов’язаних стовпцях відбувається множення значень, а це може ускладнювати інтерпретацію інших факторів, котрі мають вплив на результат.
- Принаймні один прогнозований стовпець має містити дискретизовані або дискретні значення.

У прогнозованому стовпці значення розглядаються як вхідні. Для виявлення зв'язків між стовпцями це може бути корисним.

1.5.6 Алгоритм нейронної мережі

Алгоритм нейронної мережі – це певна реалізація архітектури адаптованої та розповсюдженої нейронної мережі для машинного навчання. Алгоритм працює завдяки тестуванню кожного можливого стану вхідного атрибута з кожним можливим станом прогнозованого атрибута та застосовує навчальні дані, що розраховують ймовірності кожного поєднання. Для завдань регресії та класифікації, а також для прогнозування результату на основі вхідних атрибутів можна застосовувати ці ймовірності. Для аналізу взаємозв'язків використовується нейронна мережа [29].

За допомогою алгоритму нейронної мережі при створенні моделі інтелектуального аналізу даних можна включити декілька вихідних даних, алгоритм посприяє створенню кількох мереж. В одній моделі кількість мереж залежить від значень атрибута у вхідних стовбцях, а також від числа стовпців, які застосовуються в моделі інтелектуального аналізу даних, і числа станів у них.

Цей алгоритм корисний для аналізу саме складних даних, котрі отримуються як результат комерційних чи виробничих процесів або бізнес-задач. Для них надається пристойний обсяг навчальних даних, проте важко утворювати похідні правила з іншими алгоритмами.

Існують декілька сценаріїв використання алгоритму нейронної мережі:

- Інтелектуальний аналіз тексту;
- Аналіз таргетингу та маркетингу (перевірка ефективності рекламної кампанії на радіо, сайті чи розсилка на пошту);
- Прогнозування коливань валюти чи зміна цін;
- Аналіз виробничих та промислових процесів;
- Прогнозуюча модель, що досліджує складні зв'язки між вхідними та вихідними атрибутами, а саме їх кількістю.

Цей алгоритм створює мережу з так званими нейронами – шарами вузлів. Вони бувають вхідними, вихідними та прихованими.

- Вхідний шар – визначає значення вхідних атрибутів та їх імовірності для моделі інтелектуального аналізу даних.
- Вихідний шар – являє собою показ прогнозованих атрибутів для моделі інтелектуального аналізу даних.
- Прихований шар – отримання вхідних даних від вхідних вузлів та передача їх вихідним даним із такими ж вузлами відповідно. Вхідні атрибути у цьому випадку мають певні вагові коефіцієнти. Вони описують важливість кожного вхідного атрибуту окремо. В залежності від вагового коефіцієнту, а точніше, яку саме вагу та важливість він має, призначений атрибут набуває також більшу важливість. Також дані коефіцієнти бувають від'ємними, вони не сприяють, а навпаки перешкоджають отриманню результату.

У цій моделі повинні бути ключовий стовпець, а також вхідні та прогнозовані.

Моделі із використанням алгоритму нейронної мережі залежні від значень, заданих у доступних параметрах. Ці параметри визначають спосіб очікуваного чи звичайного розподілу даних у стовпці, а також порядок вибірки даних.

1.5.7 Алгоритм кластеризації послідовностей

Алгоритм кластеризації послідовностей – це алгоритм, що включає у себе аналіз послідовностей і кластеризації. Його застосовують для переглядання даних, котрі мають події з певною послідовністю. Алгоритм відшуковує найбільш поширені послідовності та здійснює кластеризацію для пошуку подібних послідовностей. В якості даних для машинного навчання для

отримання відомостей про бізнес-сценарії або проблеми, можна використовувати такі типи послідовностей:

- Журнали, що містять події, котрі відбулися до інциденту(збій жорсткого диска);
- Дані про відвідування та клікабельність на веб-сайтах або ж їх перегляд;
- Записи, з відстеженням взаємодії пацієнта, клієнта у конкретний час щодо надання послуг;
- Записи транзакцій із порядком додання товарів, обраних в інтернет-магазині.

Цей алгоритм має спільне із алгоритмом кластеризації. Проте відмінність полягає у тому, що він знаходить кластери варіантів саме з подібними шляхами в послідовності.

Приклад: певна компанія проводить аналіз відомостей про сторінки та їх відвідування користувачами. Через те, що клієнту перед оформленнями замовлення треба зареєструватися на сайті, компанія може слідкувати за діями в рамках вузлів, що відбувають у профілі клієнта. За допомогою цього можна знайти за схожими послідовностями натискань та діями такі ж групи або кластери клієнтів, цьому і сприяє алгоритм кластеризації послідовностей. Також ці кластери використовуються компанією для визначення сторінок , на які клієнт може перейти далі та якими можуть бути його наступні дії на конкретному веб-сайті.

Особливість алгоритму кластеризації полягає у використанні даних послідовностей. Ці дані являють собою переходи між станами в наборі або ряд подій. У цьому випадку це ланцюг придбання товарів або маніпуляції із натисканням миші на веб-сайтах та сторінках. Алгоритм досліджує відмінності та ймовірність переходів або відстані між усіма послідовностями в наборі даних. Це відбувається з метою визначення, які з послідовностей доцільно застосовувати в якості вхідних даних для кластеризації. Після того, як алгоритм

створив списки ймовірних послідовностей, відбувається застосування цих відомостей у якості вхідних даних для кластеризації за допомогою метода максимізації очікувань (EM).

Під час підготовки даних, які призначені для кластеризації послідовностей, варто враховувати вимоги до обраного алгоритму, а також і до обсягу необхідних даних та їх використання.

Вимоги до моделі кластеризації послідовностей:

- Стовпець ідентифікатора послідовності. Тип даних, що включає ідентифікатор послідовності, може виглядати по-різному. Наприклад, може застосовувати ціле число, веб-сторінки чи рядок із текстом при умові, що стовпець розрізняє події в послідовності. Для кожної із послідовностей є допустимим тільки один код послідовності, а кожна модель допускає єдиний тип послідовності.
- Одиночний ключовий стовпець. Модель кластеризації повинна мати ключ для ідентифікації записів.
- Необов'язкові атрибути, що не включені до послідовності. Наприклад, вкладені стовпці. Алгоритм підтримує різноманітні атрибути.

Наприклад, розглядаючи ситуацію із діяльністю клієнта у інтернет-магазині, модель кластеризації послідовності може враховувати не пов'язані з послідовністю атрибути: таблиця варіантів, демографічні та дані про замовлення клієнта. До того ж, вона міститиме таблицю з послідовністю здійснення покупок або перегляду веб-сайту клієнтом в якості даних послідовності [30].

1.5.8 Алгоритм часових рядів

Цей алгоритм надає кілька алгоритмів, що оптимізовані для прогнозування безперервних значень, таких як продажі товарів у часі. Модель часових рядів не потребує додаткових полів нових відомостей для

прогнозування тренду, як цього потребує дерево прийняття рішень. Спрогнозувати тенденції можна на основі вихідного набору даних, що використовується для побудови моделі. Під час прогнозування є можливість вводити нові дані в модель і вони будуть задіяні при аналізі тенденцій автоматично.

У наступній діаграмі (рис. 1.17) демонструється типова модель прогнозування. Там показано продажі продуктів у чотирьох відмінних регіонах протягом певного часу. Продажі у кожному з регіонів регіоні виділяються жовтою, червоною, бузковою та синьою лініями, а для кожного регіону лінія має дві частини.

- Зображення історичних даних – зліва від вертикальної лінії – дані, що використовуються для створення моделі.
- Прогнозовані дані – праворуч від вертикальної лінії – прогноз, що зроблений моделлю.

  Ряд – поєднання вихідних даних і прогнозованих даних.

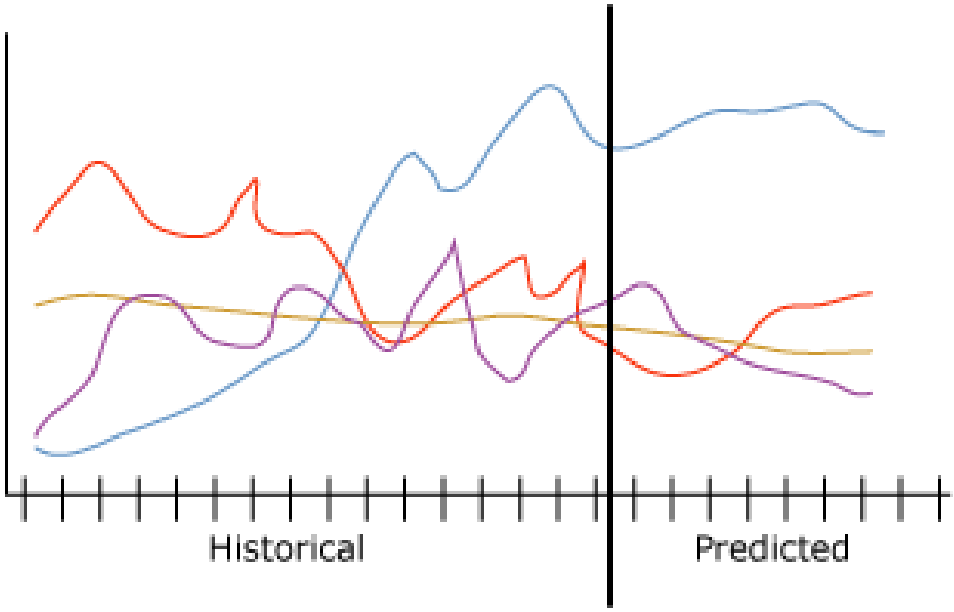


Рисунок 1.17 – Модель прогнозування з застосуванням алгоритму часових рядів

Виконання перехресного прогнозу – важлива характеристика алгоритму часових рядів. Підсумкова модель при навчанні алгоритму двома пов'язаними або окремими рядами використовується для прогнозування результату одного ряду, що базується на поведінці іншого. Наприклад, спостереження за продажами одного продукту може мати вплив на прогнозований інший.

Застосування перехресних прогнозів також корисне, щоб створити загальну модель, що застосовується для кількох рядів. Наприклад, прогнози для конкретного регіону нестабільні через відсутність даних хорошої якості. Загальна модель навчається на середньому арифметичному всіх чотирьох регіонів, а потім її можна застосувати до окремих рядів для підготовки стабільніших прогнозів по конкретному регіону.

Приклад: керівництву компанії потрібне прогнозування щомісячних продажів велосипедів на наступний рік. Більше цікавить використання продажу однієї моделі велосипедів для прогнозу продажів іншої. Модель інтелектуального аналізу даних, що прогнозує продажу велосипедів у майбутньому можна отримати шляхом застосування алгоритму часових рядів. Також можна скористатися перехресним прогнозуванням для цього.

Компанія має на меті поповнювати модель свіжими даними про продажі й оновлювати прогнози щокварталу для моделювання останніх тенденцій. Для внесення правок магазинам, що нерегулярно оновлюють дані, створили модель прогнозування, що є загальною для всіх регіонів.

Принцип роботи алгоритму: ARTXP був єдиним, що застосовувався в SQL Server 2005. Алгоритм ARTXP розроблявся для короткострокових прогнозів, тому наступне ймовірне значення прогнозувалося у ряді. Другий алгоритм ARIMA розраховувався для довгострокового прогнозування і був доданий з версії SQL Server 2008.

За замовчуванням в алгоритмі часових рядів застосовується комбінація алгоритмів для аналізування закономірностей і підготовки прогнозів є в

алгоритмі за замовчуванням. Вивчення моделі відбувається на тих же даних: в одній застосовується ARTXP, а в іншій – алгоритм ARIMA. Далі для сформування найкращого прогнозу для змінного числа часових зрізів, відбувається поєднання результатів двох моделей. Алгоритм ARTXP більш надійний, тому що підходить для короткострокових прогнозів. Проте алгоритм ARIMA при цьому стає кориснішим через те, що часові зрізи йдуть вперед.

Процесом поєднання алгоритмів можна керувати, для цього варто переносити акцент у часових рядах на довгостроковий або короткостроковий. Використання алгоритму, що можна вказати ще з версії SQL Server 2008 Standard:

- ARIMA – лише для довгострокового прогнозування.
- ARTXP – тільки для короткострокового прогнозування.
- Поєднання обох алгоритмів – за замовчуванням.

Можна налаштувати спосіб об'єднання моделей прогнозування в даному алгоритмі, починаючи з версії SQL Server 2008 Enterprise. Алгоритм часових рядів, при застосуванні змішаної моделі, об'єднує два алгоритми таким чином:

- Алгоритм ARTXP – для формування тільки двох перших;
- ARIMA і ARTXP – після прогнозування двох перших використовують дані алгоритми;
- Далі частка алгоритму ARIMA в прогнозуванні відмовляється від застосування ARTXP;
- Керувати швидкістю зниження частки алгоритму ARTXP і підвищення цього у ARIMA, а також моментом поєднання алгоритмів, налаштовуючи параметр PREDICTION_SMOOTHING.

Відслідкувати вплив сезонних чинників на дані на кількох рівнях можна за допомогою обох алгоритмів. Наприклад, посеред циклу річних змін даних можуть існувати місячні. Для визначення вводяться дані про періодичність або задається автоматичний режим.

Під час підготовки вибірки даних для застосування в навчанні будь-якої моделі Data mining варто знати певні вимоги конкретної моделі і способи використання даних.

От основні вимоги:

- Прогнозований стовпець допомагає створити модель часових рядів. Він має містити тип даних із безперервними значеннями. Наприклад, можна прорахувати зміну атрибутів (обсяг продажів, дохід чи температура). Не допускається застосування дискретних даних у якості прогнозованого стовпця (споживчий статус або рівень освіти).
- Окремий ключовий стовпець часу – тип date або числовий. Використовується в якості набору варіантів для визначення , що часових зрізів, що використовує модель. Типом даних стовпця «key time» може бути числовий тип або datetime. Варто пам’ятати, що у стовпці мають бути безперервні, унікальні для кожного ряду, значення. Набір варіантів для моделі не може бути в двох стовпчиках, наприклад, «Місяць» або «Рік».
- Необов'язковий ключовий стовпець ряду має містити особливі значення для ідентифікації ряду, вони повинні бути унікальними. Наприклад, при існуванні лише одного запису для кожної назви товару в часовому зрізі, одна модель може включати дані про продажі багатьох моделей продуктів.

1.6 Висновки

У даному розділі було розглянуто теоретичні відомості по заданій темі. Визначено поняття серверу БД MS SQL Server та середовища розробки Microsoft Visual Studio. Розглянуті найрозповсюдженіші алгоритми, які надаються даними сервісами.

Було вирішено задачі, на основі який будуть розглядатись алгоритми. В якості основних задач для тестувань було обрано наступні алгоритми:

- Алгоритм кластеризації
- Алгоритм дерева прийняття рішень
- Алгоритм лінійної регресії
- Алгоритм взаємозв’язків
- Спрощений алгоритм Байеса
- Алгоритм нейронної мережі
- Алгоритм кластеризації послідовностей
- Алгоритм часових рядів

Для таких задач:

- Прогнозування дискретного атрибуту
- Прогнозування безперервного атрибуту
- Прогнозування послідовності
- Знаходження груп спільних елементів в транзакціях

Особливу увагу потрібно приділити оцінки ефективності, тобто наскільки кожен алгоритм буде кращий для конкретної задачі.

2 РЕАЛІЗАЦІЯ ТА АНАЛІЗ ЕФЕКТИВНОСТІ РЕАЛІЗАЦІЇ АЛГОРИТМІВ DATA MINING

2.1 Вступ

В інтелектуальному аналізі даних(Analysis Services) алгоритм означає набір обчислень і евристики. Він і створює модель на основі даних. Для цього спочатку триває аналіз даних за допомогою пошуку певних тенденцій та закономірностей. Щоб підібрати оптимальні параметри для створення моделі, алгоритм використовує результати аналізу до безлічі ітерацій. Наступним кроком є застосування цих параметрів до усього набору, це дозволяє отримати докладну статистику та виявити закономірності, що є придатними для використання.

Модель, що створена алгоритмом з отриманих даних набуває таких форм:

- Математична модель, котра прогнозує продаж;
- Набір кластерів, що описують зв'язки варіантів;
- Дерево рішень, яке передбачає результат і оцінює вплив різних критеріїв на нього;
- Групування продуктів в транзакції та ймовірність одночасної покупки продуктів.

SQL Server використовує вивчені та найпопулярніші методи виявлення закономірностей в даних. Наприклад, алгоритм кластеризації методом К-середніх – один із найстаріших алгоритмів, його застосовують у багатьох інструментах та з різними параметрами. Даний алгоритм, що реалізований в інтелектуальному аналізі даних SQL Server, був розроблений групою Microsoft Research, а потім його оптимізували для роботи з Analysis Services. Усі алгоритми Майкрософт доступні для налаштування та програмування з використанням наданих API. Також можна також автоматизувати навчання та створення моделей за допомогою компонентів інтелектуального аналізу даних у службах Integration Services.

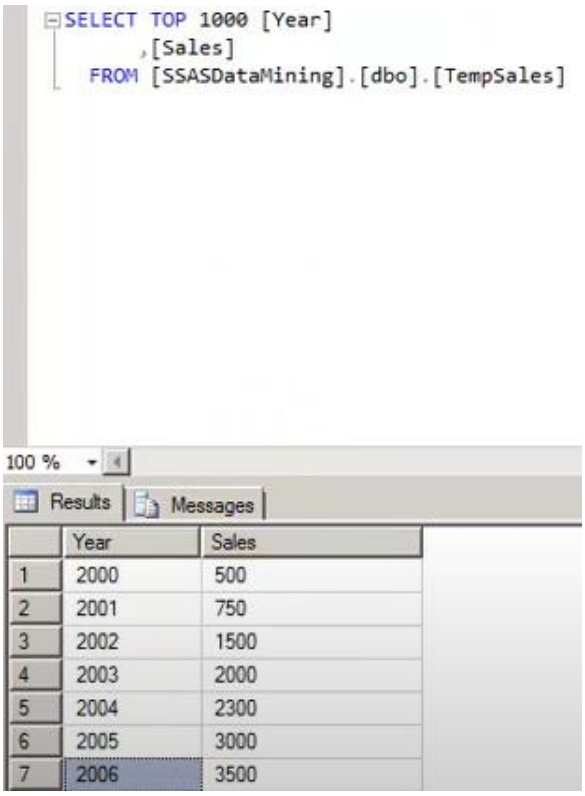
До того ж підтримується і використання OLE DB. Можна також розробляти власні алгоритми, а, зареєструвавши їх в якості служб, потім використовувати на платформі SQL Server.

2.2 Задача прогнозування неперервного атрибуту

В даному розділі буде розглядатись алгоритм часових рядів для вирішення задачі прогнозування продажів на наступний рік та алгоритм дерева прийняття рішень та алгоритм лінійної регресії для вирішення задачі оцінки ризику з врахуванням демографії.

2.2.1 Алгоритм часових рядів

Для початку необхідно створити початкову вибірку у вигляді таблиці, як показано на рис. 2.1.



	Year	Sales
1	2000	500
2	2001	750
3	2002	1500
4	2003	2000
5	2004	2300
6	2005	3000
7	2006	3500

Рисунок 2.1 – Вибірка для алгоритму часових рядів

Після цього потрібно створити проект, що продемонстровано, на наступному рис. 2.2.

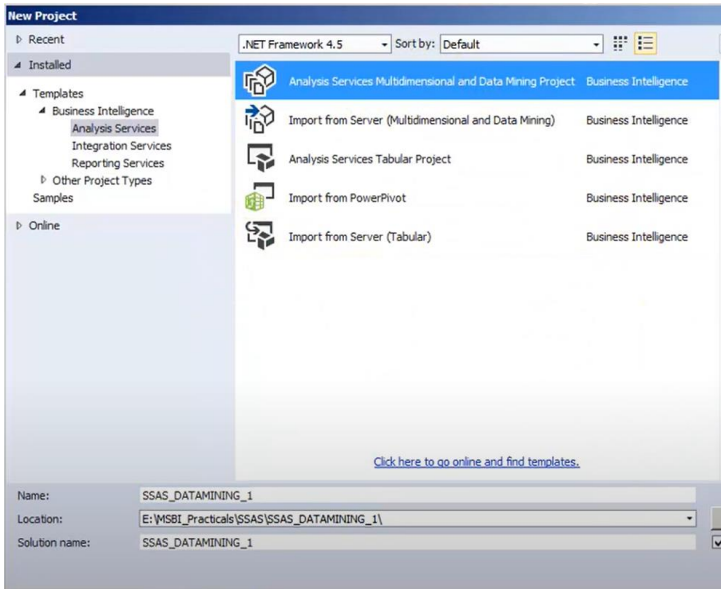


Рисунок 2.2 – Створення проекту

Наступним кроком є створення джерела даних. Для цього потрібно підключитись до БД (рис. 2.3).

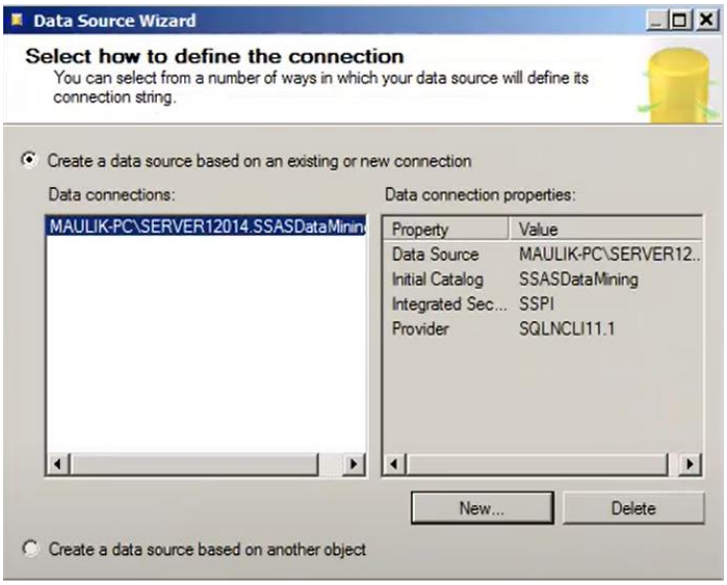


Рисунок 2.3 – Підключення до БД

Після цього необхідно створити уявлення джерела даних, щоб підключити таблицю з вибіркою даних (рис. 2.4).

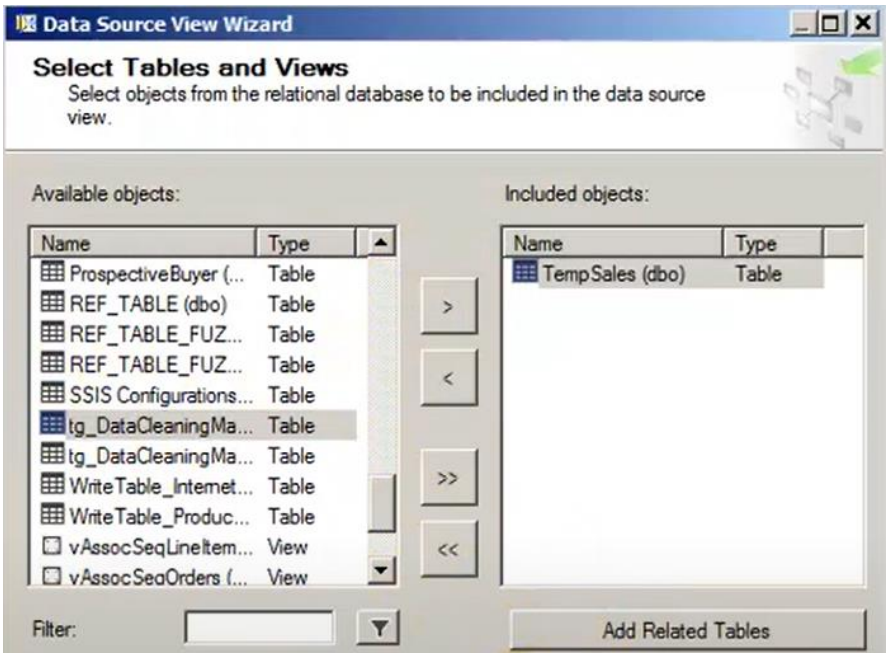


Рисунок 2.4 – Вибір таблиці для побудування моделі

Наступним кроком є створення структури видобутку даних, де необхідно обрати алгоритм часових рядів (рис. 2.5).

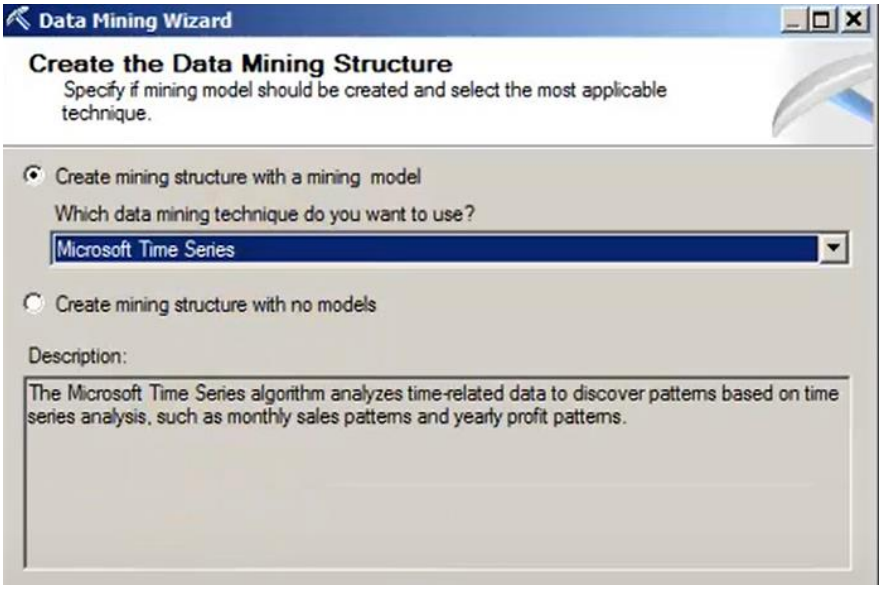


Рисунок 2.5 – Вибір алгоритму часових рядів

Далі потрібно вибрати вхідні, ключові поля і поле, на основі якого буде будуватися прогноз (рис. 2.6). Для цього прикладу вхідний і прогнозоване поля мають однакову роль.

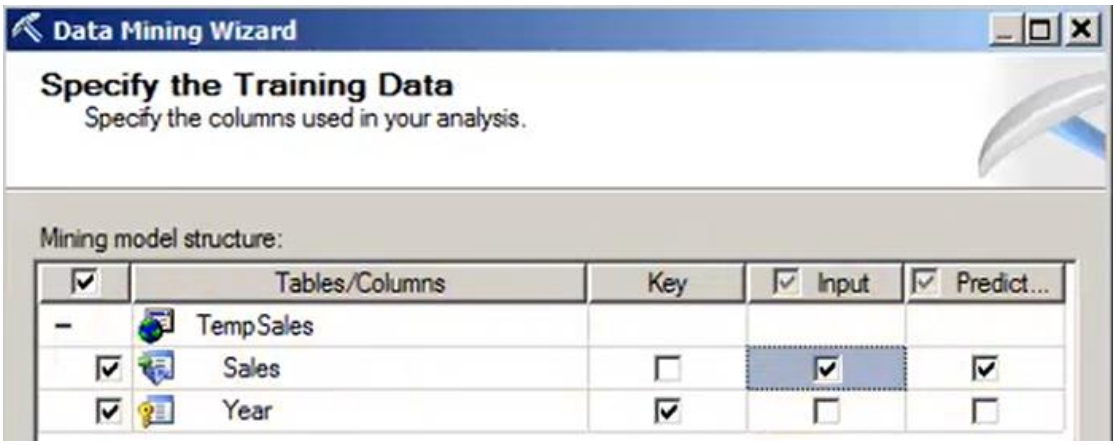


Рисунок 2.6 – Вибір вхідних, ключових та вихідних полів для алгоритму часових рядів

Після цього кроку можна спостерігати пропоновану структуру для видобутку даних (рис. 2.7).

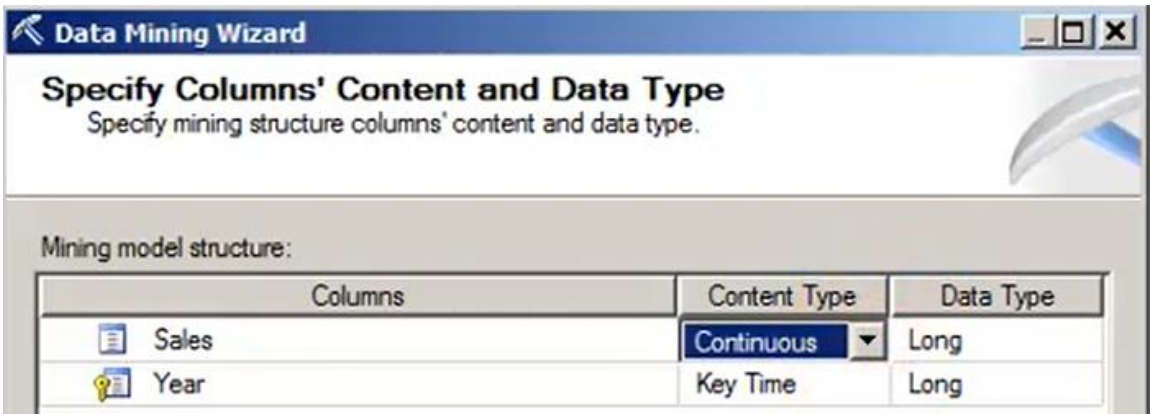


Рисунок 2.7 - Підтвердження запропонованої моделі

Наступним кроком є успішне розгортання проекту (рис. 2.8).

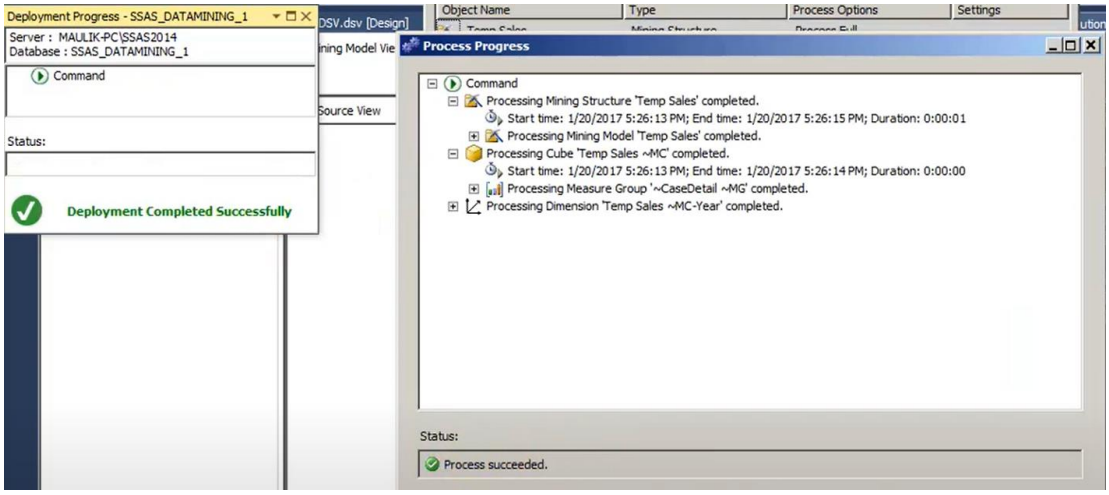


Рисунок 2.8 – Успішне розгортання проекту для алгоритму часових рядів

Побудовану модель можна знайти на вкладці з уявлення моделі видобутку даних (рис. 2.9). В ній можна указати глибину прогнозування (в даному прикладі це значення становить «5»).

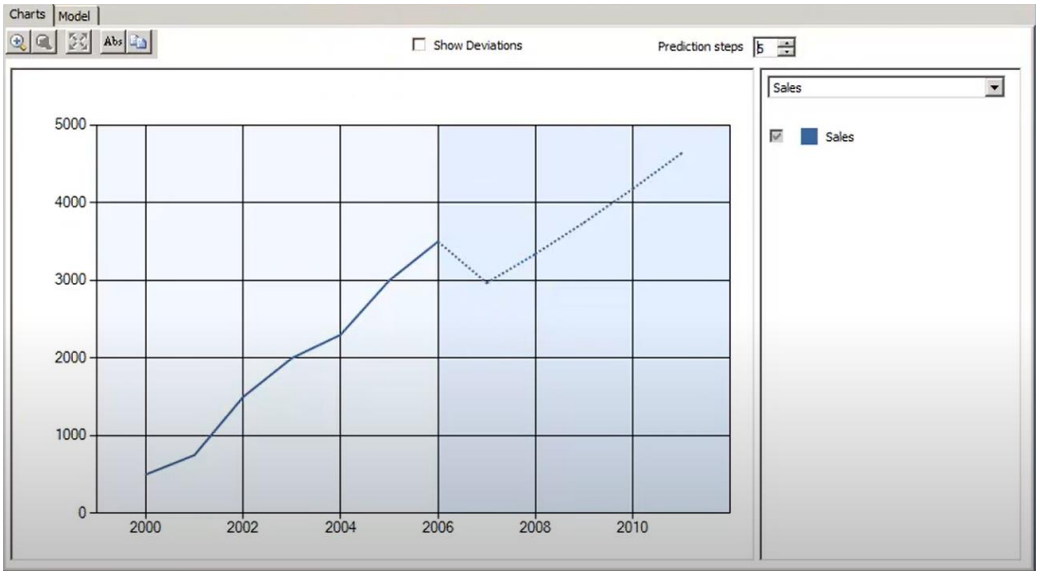


Рисунок 2.9 – Побудована модель для алгоритму часових рядів

Також можна вивести результат у вигляді таблиці в самому MS SQL Server (рис. 2.10).

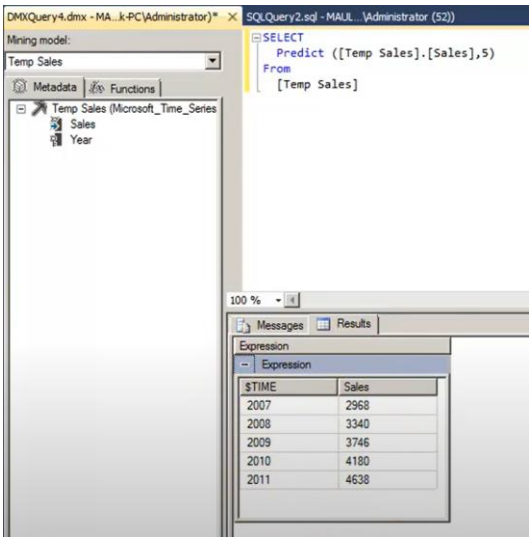


Рисунок 2.10 – Виклик моделі з використанням алгоритму часових рядів в MS SQL Server

2.2.2 Алгоритм логістичної регресії та алгоритм дерева прийняття рішень

В даному розділі застосовується так ж сама вибірка, що і в попередньому розділі, але прогнозованим атрибутом буде не прапор покупки велосипеду, а річний дохід з продажів (рис. 2.11). Проте алгоритм лінійної регресії підтримує лише поля з неперервними значеннями.

Structure	REGRESSION_TREES	LINEAR_REGRESSION
	Microsoft_Decision_Trees	Microsoft_Linear_Regression
Age	Input	Input
Bike Buyer	Input	Ignore
Commute Distance	Input	Ignore
Customer Key	Key	Key
English Education	Input	Ignore
English Occupation	Input	Ignore
Gender	Input	Ignore
House Owner Flag	Input	Ignore
Marital Status	Input	Ignore
Number Cars Owned	Input	Input
Number Children At Home	Input	Ignore
Region	Input	Ignore
Total Children	Input	Ignore
Yearly Income	PredictOnly	PredictOnly

Рисунок 2.11 - Вибір вхідних, ключових та вихідних даних для алгоритму дерева прийняття рішень та алгоритму лінійної регресії

Після застосування алгоритму дерева прийняття рішень можна отримати таку статистику (рис. 2.12).

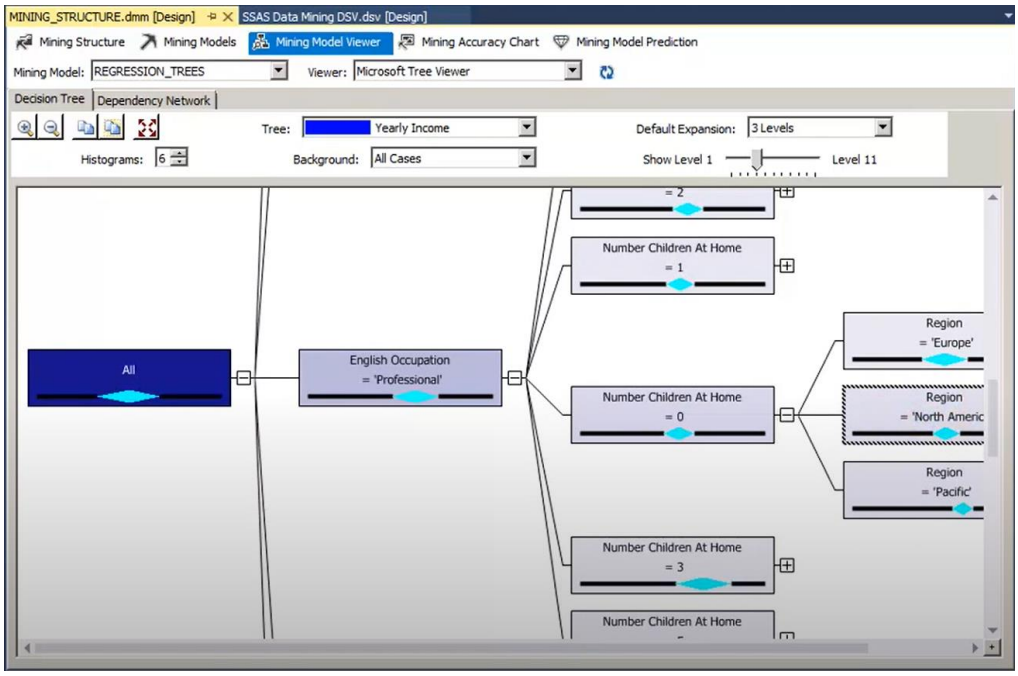


Рисунок 2.12 – Приклад моделі дерева прийняття рішень

Даний алгоритм розраховує формулу, по якій можна розрахувати річний прибуток з продажів велосипедів в залежності від віку клієнта і кількості машин (рис. 2.13).

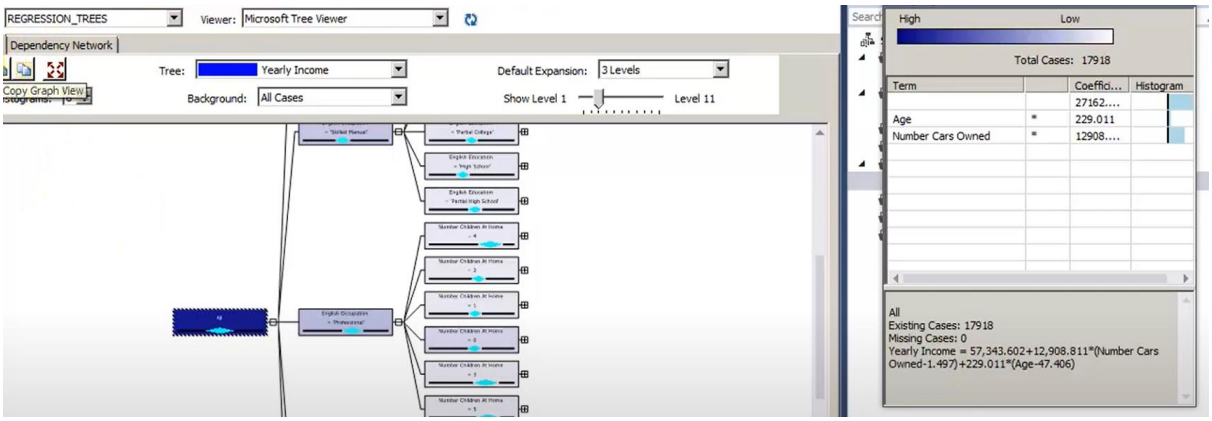


Рисунок 2.13 – Розрахунок загальної формули прибутку з використанням алгоритму дерева прийняття рішень

В даному алгоритмі наявна функція, яка перераховує формулу річного прибутку, в залежності від того, який вузол дерева було обрано (рис. 2.14).

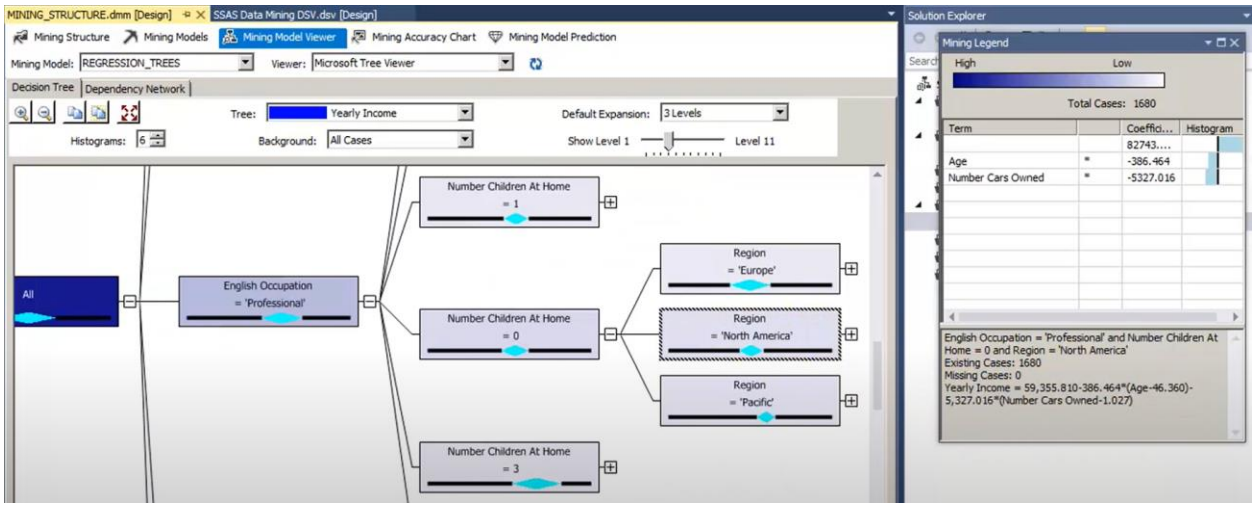


Рисунок 2.14 – Розрахунок формули прибутку для окремо взятого вузла дерева з використанням алгоритму дерева прийняття рішень

Для алгоритму лінійної регресії можна отримати лише загальну формулу (рис. 2.15).

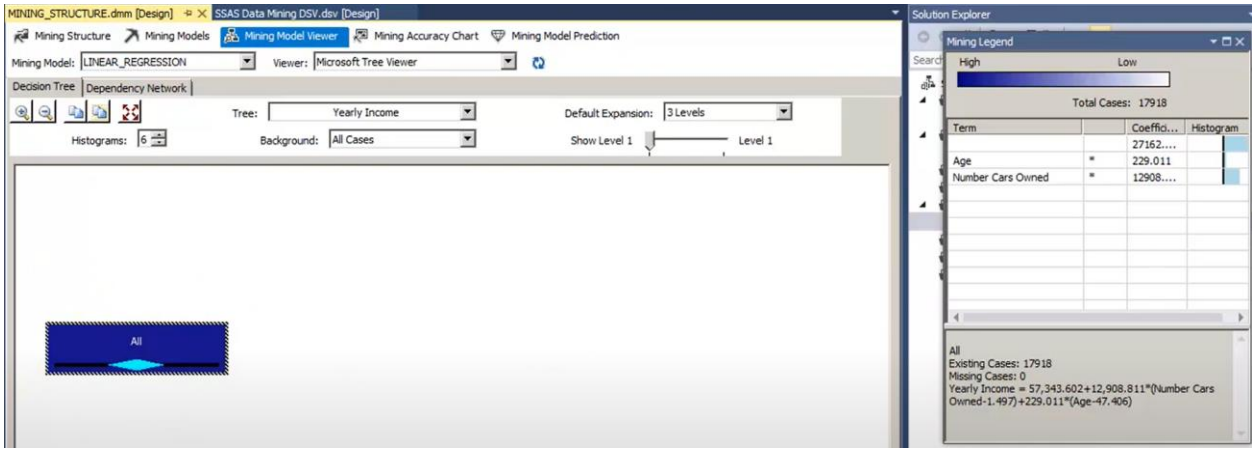


Рисунок 2.15 - Розрахунок загальної формули прибутку з використанням алгоритму лінійної регресії

В результаті роботи обох алгоритмів було отримано ідентичну формулу для прибутку.

2.2.3 Порівняльний аналіз

Отже, після реалізації та дослідження задачі прогнозування неперервного атрибуту з використанням алгоритму часових рядів, алгоритму лінійної регресії та алгоритму дерева прийняття рішень можна виділити два типу задач:

- Прогноз продаж на наступний рік
- Формування оцінки ризику з врахуванням демографії

Для першого варіанту кращим вибором буде алгоритм часових рядів, оскільки інші алгоритми (алгоритм лінійної регресії та алгоритм дерева прийняття рішень) неможливо застосувати до даного типу задач.

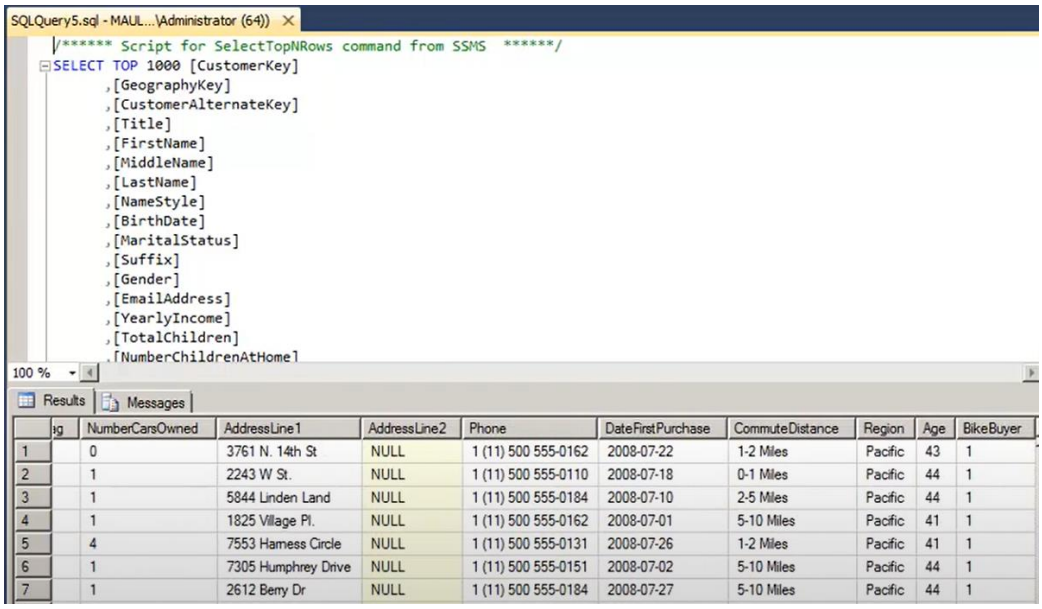
Щодо другого типу – кращим вибором буде алгоритм дерева прийняття рішень, оскільки після використання алгоритму лінійної регресії та алгоритму дерева прийняття рішень до однієї вибірки даних було отримано прогноз продажів з однаковою точністю (див. рис. 2.13 та рис. 2.15). Проте, відмінністю цих алгоритмів полягає в більшій гнучкості алгоритму дерева прийняття рішень, оскільки при застосуванні алгоритму лінійної регресії було отримано лише формулу (див. рис. 2.15), а при застосуванні алгоритму дерева прийняття рішень можна окремо переглянути формулу прогнозу продажів на кожній вітці дерева (див. рис. 2.14).

2.3 Задача прогнозування дискретного атрибуту

В даному розділі буде розглянуто вирішення задачі помітки клієнта із списку потенційних покупців як хороших і поганих кандидатів для алгоритму дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі та продемонстровано спрощений алгоритм Байеса.

2.3.1 Алгоритм дерева прийняття рішень, алгоритм кластеризації та алгоритм нейронної мережі

Для алгоритму дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі в роботі буде використана наступна таблиця з особистими даними клієнтів та прапором, який свідчить про те – чи купляв клієнт велосипед (проставляється значення «1» у випадку покупки) (рис. 2.16).



	ig	NumberCarsOwned	AddressLine1	AddressLine2	Phone	DateFirstPurchase	CommuteDistance	Region	Age	BikeBuyer
1		0	3761 N. 14th St	NULL	1 (11) 500 555-0162	2008-07-22	1-2 Miles	Pacific	43	1
2		1	2243 W St.	NULL	1 (11) 500 555-0110	2008-07-18	0-1 Miles	Pacific	44	1
3		1	5844 Linden Land	NULL	1 (11) 500 555-0184	2008-07-10	2-5 Miles	Pacific	44	1
4		1	1825 Village Pl.	NULL	1 (11) 500 555-0162	2008-07-01	5-10 Miles	Pacific	41	1
5		4	7553 Harness Circle	NULL	1 (11) 500 555-0131	2008-07-26	1-2 Miles	Pacific	41	1
6		1	7305 Humphrey Drive	NULL	1 (11) 500 555-0151	2008-07-02	5-10 Miles	Pacific	44	1
7		1	2612 Berry Dr	NULL	1 (11) 500 555-0184	2008-07-27	5-10 Miles	Pacific	44	1

Рисунок 2.16 – Вибірка для алгоритмів дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі

Надалі необхідно виконати ті ж самі кроки, що описані в попередньому алгоритмі до створення структури видобутку даних. На цьому кроці потрібно обрати алгоритм дерева прийняття рішень (рис. 2.17).

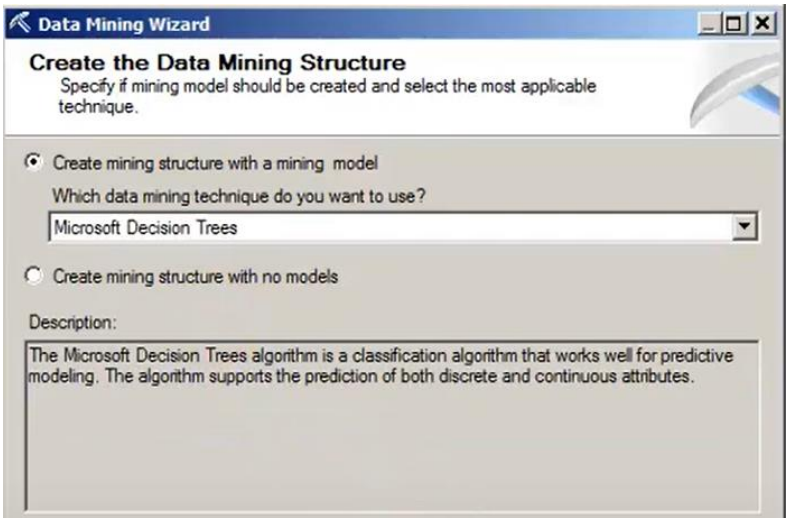


Рисунок 2.17 – Вибір алгоритму дерева прийняття рішень

В наступному кроці обов'язково потрібно обрати в якості ключового поля – унікальний ідентифікатор клієнта; в якості прогнозуючого поля – прапор, що свідчить про покупку велосипеда; в якості вхідних полів – особисті дані клієнта, по яким наявна деяка концентрація і по яким можна побудувати статистику (рис. 2.18).

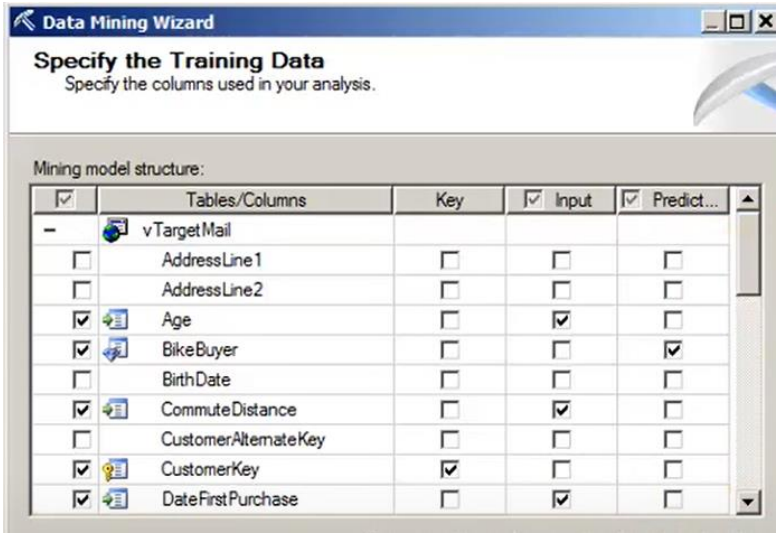


Рисунок 2.18 - Вибір вхідних, ключових та вихідних полів для алгоритму дерева прийняття рішень

Після цього будується наступна модель видобутку даних, яка показана на рис. 2.19.

Structure	DECISION_TREE
	Microsoft_Decision_Trees
Age	Input
Bike Buyer	PredictOnly
Commute Distance	Input
Customer Key	Key
Date First Purchase	Input
English Education	Input
English Occupation	Input
Gender	Input
House Owner Flag	Input
Marital Status	Input
Number Cars Owned	Input
Number Children At Home	Input
Region	Input
Spanish Occupation	Input
Total Children	Input
Yearly Income	Input

Рисунок 2.19 – Побудована модель для алгоритму дерева прийняття рішень

До моделі необхідно додати ще 2 алгоритми – алгоритм кластеризації та алгоритм нейронної мережі як показано на рис. 2.20 – 2.21.

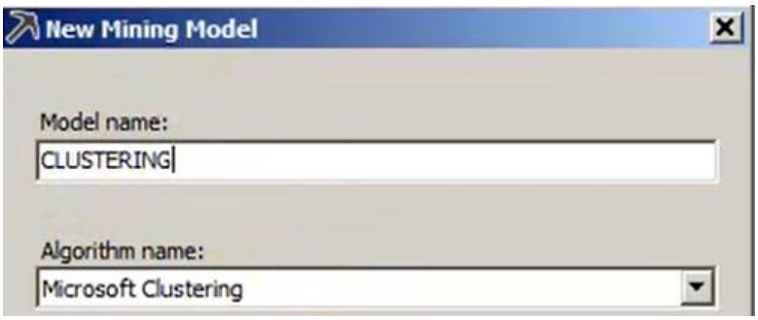


Рисунок 2.20 – Додання до моделі алгоритму кластеризації

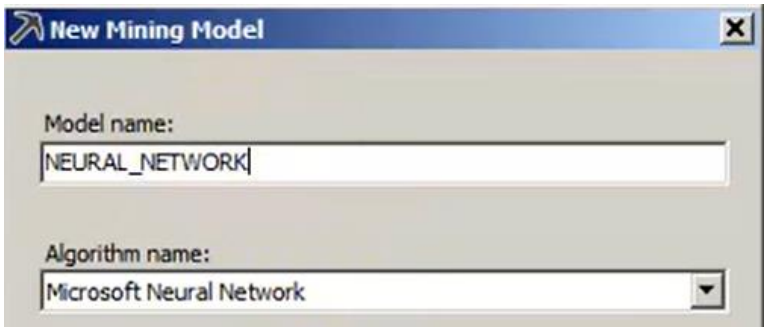


Рисунок 2.21 – Додання до моделі алгоритму нейронної мережі

Після додання вказаних алгоритмів модель набуває наступного вигляду, що продемонстровано на рис. 2.22.

Structure	DECISION_TREE	CLUSTERING	NEURAL_NETWORK
	Microsoft_Decision_Trees	Microsoft_Clustering	Microsoft_Neural_Network
Age	Input	Input	Input
Bike Buyer	PredictOnly	PredictOnly	PredictOnly
Commute Distance	Input	Input	Input
Customer Key	Key	Key	Key
Date First Purchase	Input	Input	Input
English Education	Input	Input	Input
English Occupation	Input	Input	Input
Gender	Input	Input	Input
House Owner Flag	Input	Input	Input
Marital Status	Input	Input	Input
Number Cars Owned	Input	Input	Input
Number Children At Home	Input	Input	Input
Region	Input	Input	Input
Spanish Occupation	Input	Input	Input
Total Children	Input	Input	Input
Yearly Income	Input	Input	Input

Рисунок 2.22 – Побудована модель для алгоритмів дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі

Наступним кроком є успішне розгортання проекту (рис. 2.23).

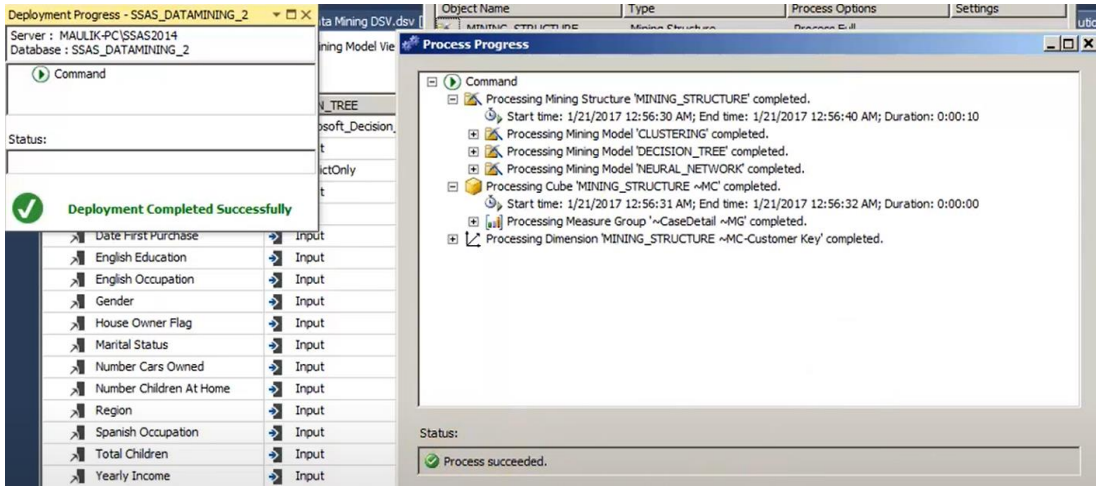


Рисунок 2.23 – Успішне розгорнення проекту для алгоритмів дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі

Якщо застосувати до моделі алгоритм дерева прийняття рішень, то можна спостерігати наступну картину, зображену на рис. 2.24. Чим темніший прямокутник, який відповідає значенням конкретній змінній, тим вона краще розподіляє вибірку.

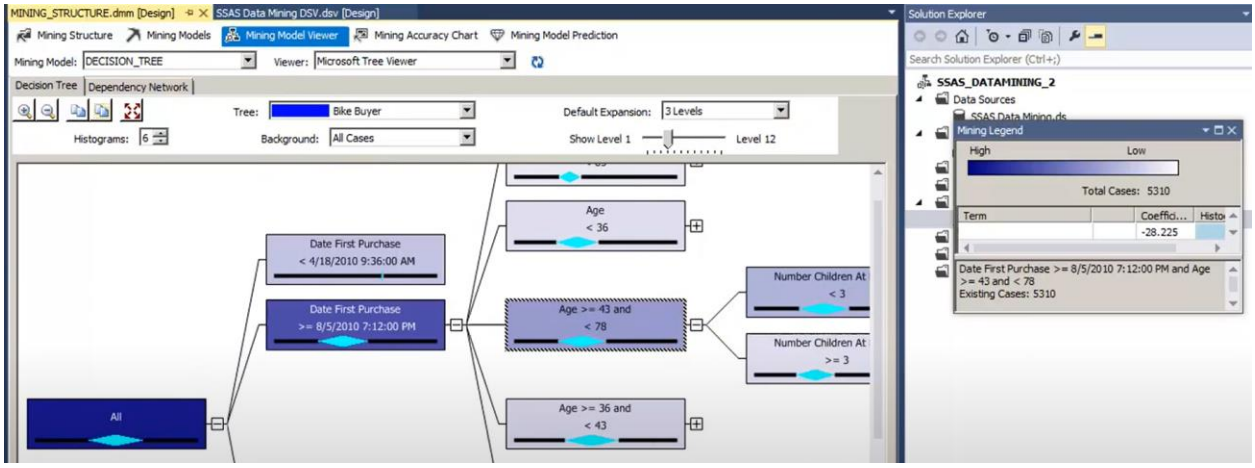


Рисунок 2.24 – Модель х застосуванням алгоритму дерева прийняття рішень

Частиною аналітики даного алгоритму також є наступна картина (рис. 2.25 - 2.26), яка демонструє «найсильніші» зміні, які найкраще впливають на ймовірність покупки велосипеду. На даному прикладі можна сказати, що змінна «Дата першої покупки» - найсильніша, а «стать» - найслабша.

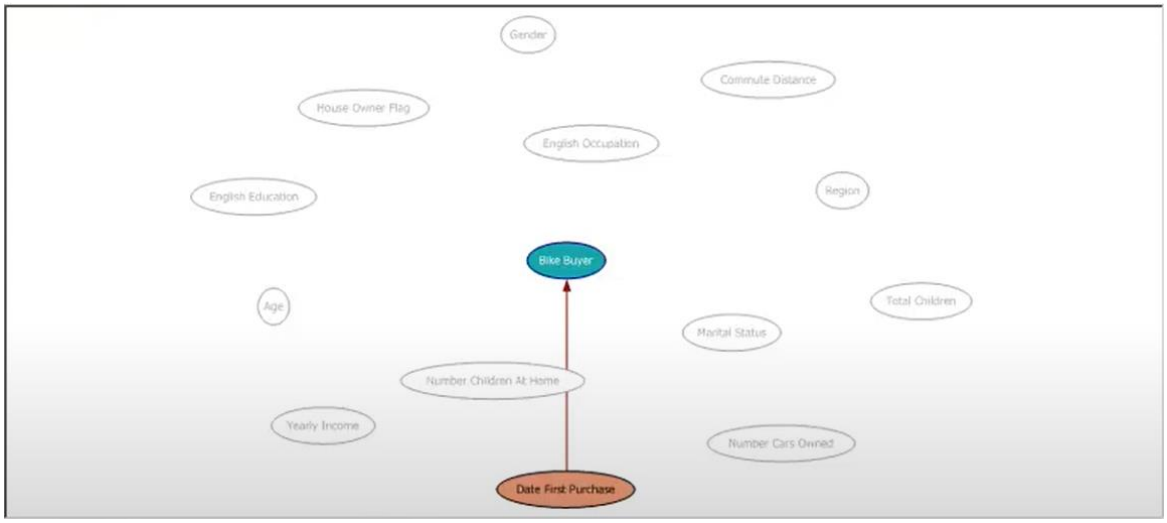


Рисунок 2.25 – «Найсильніша» зміна для алгоритму дерева прийняття рішень

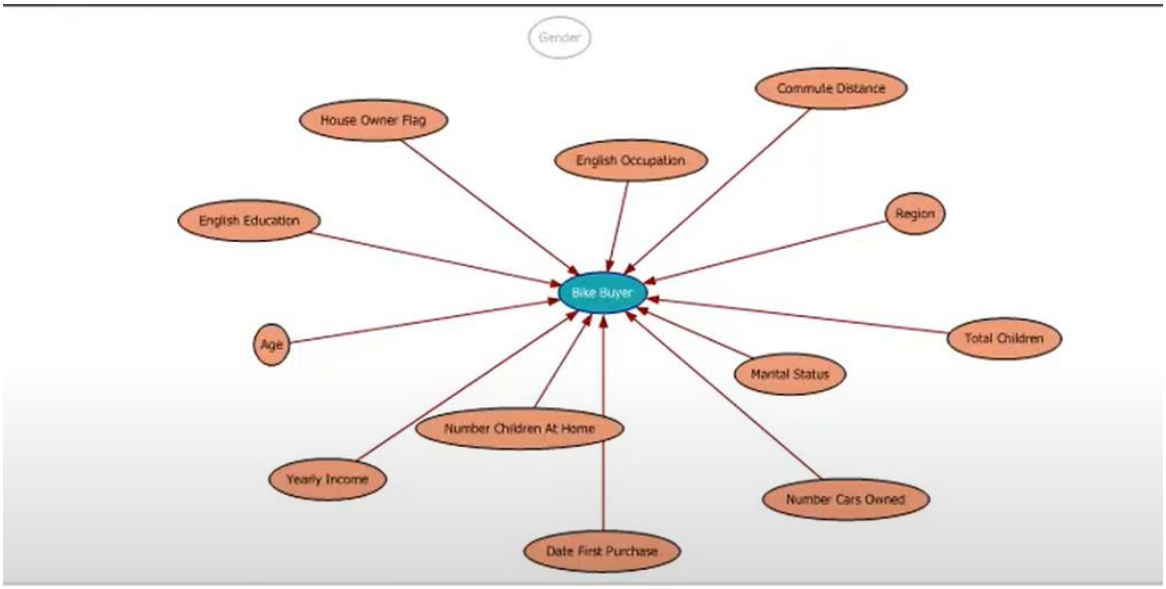


Рисунок 2.26 – «Найслабша» зміна для алгоритму дерева прийняття рішень

Якщо ж застосувати алгоритм кластеризації, то модель буде виглядати наступним чином (рис. 2.27). Можна спостерігати за яким принципом була здійснена розбивка на кластери.

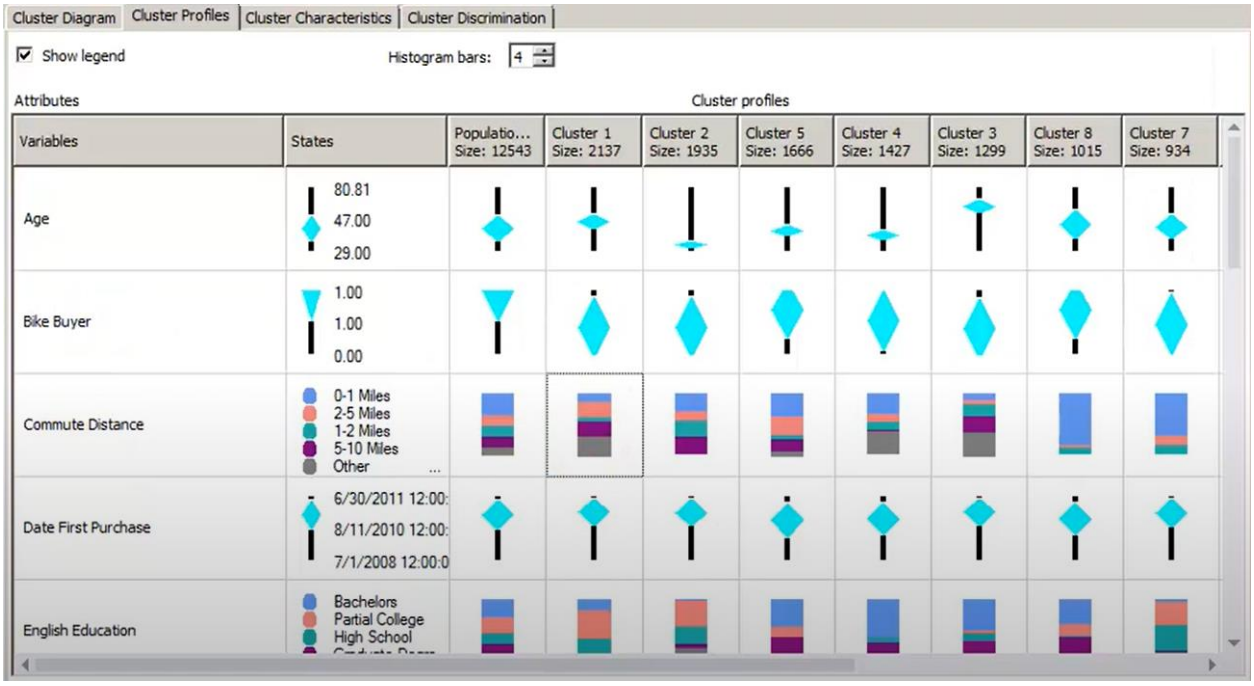


Рисунок 2.27 – Модель з застосуванням алгоритму кластеризації

При застосуванні до моделі алгоритму нейронної мережі можна спостерігати наступну картину (рис. 2.28). В даному алгоритмі наявна можливість розрахувати можливість покупки велосипеду в залежності від значення змінної. Наприклад, для віку покупця від 29 до 39.752 вірогідність шансу покупки велосипеду від 0.0843 до 1 становить 65.07%, а для шансу покупки велосипеду від 0 до 0.168 становить 2.02%.

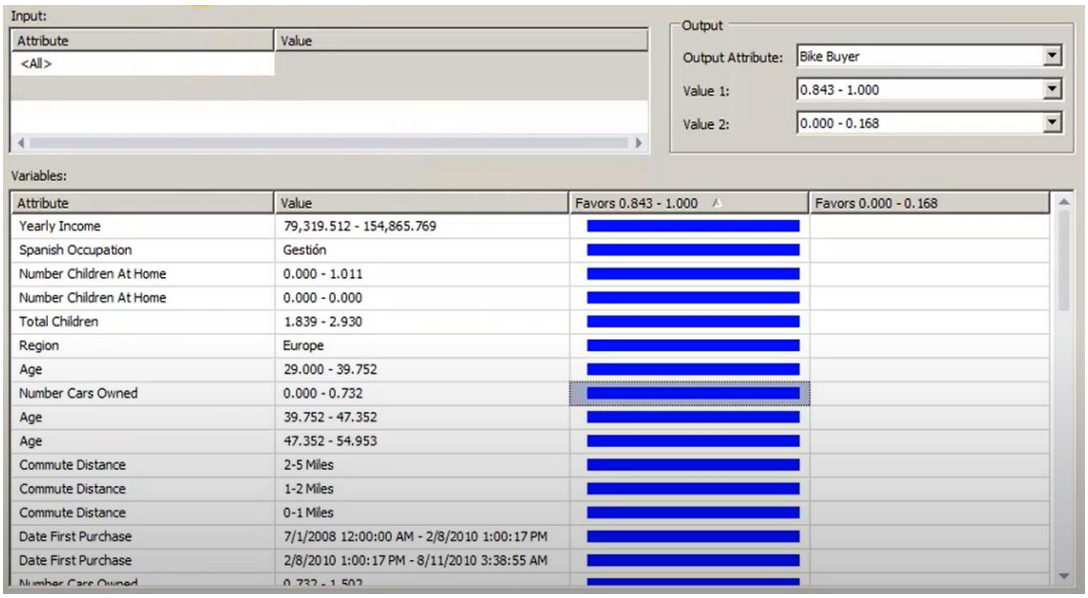


Рисунок 2.28 – Модель з використанням алгоритму нейронної мережі

При застосуванні до моделі різних алгоритмів можна застосувати функцію, яка розрахує найкращий алгоритм, який варто застосувати для побудови аналітики для даної вибірки (рис. 2.29).

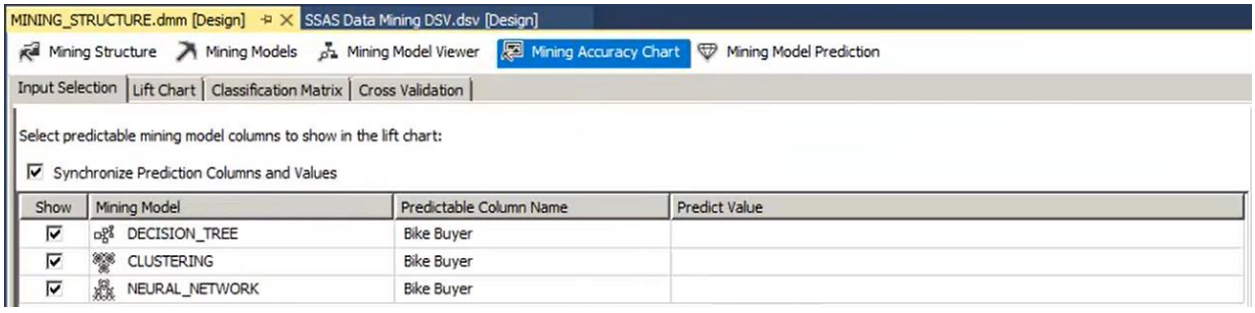


Рисунок 2.29 – Вибір моделей для порівняння

І найкращим вибором для даної вибірки буде алгоритм нейронних мереж, а найгіршим – алгоритм дерева прийняття рішень (рис. 2.30). Дана функція призначає бал для обраних алгоритмів від 0 до 1.

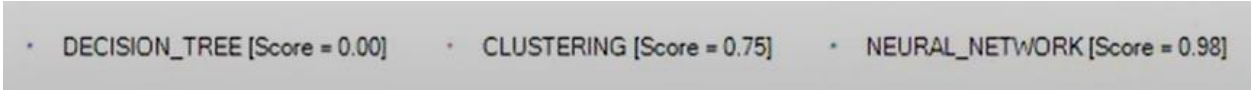


Рисунок 2.30 – Результати порівняння алгоритмів дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі

2.3.2 Спрощений алгоритм Байеса

Для даного алгоритму була застосована та сама вибірка, що і для попереднього розділу, але з іншими вхідними даними (рис. 2.31).

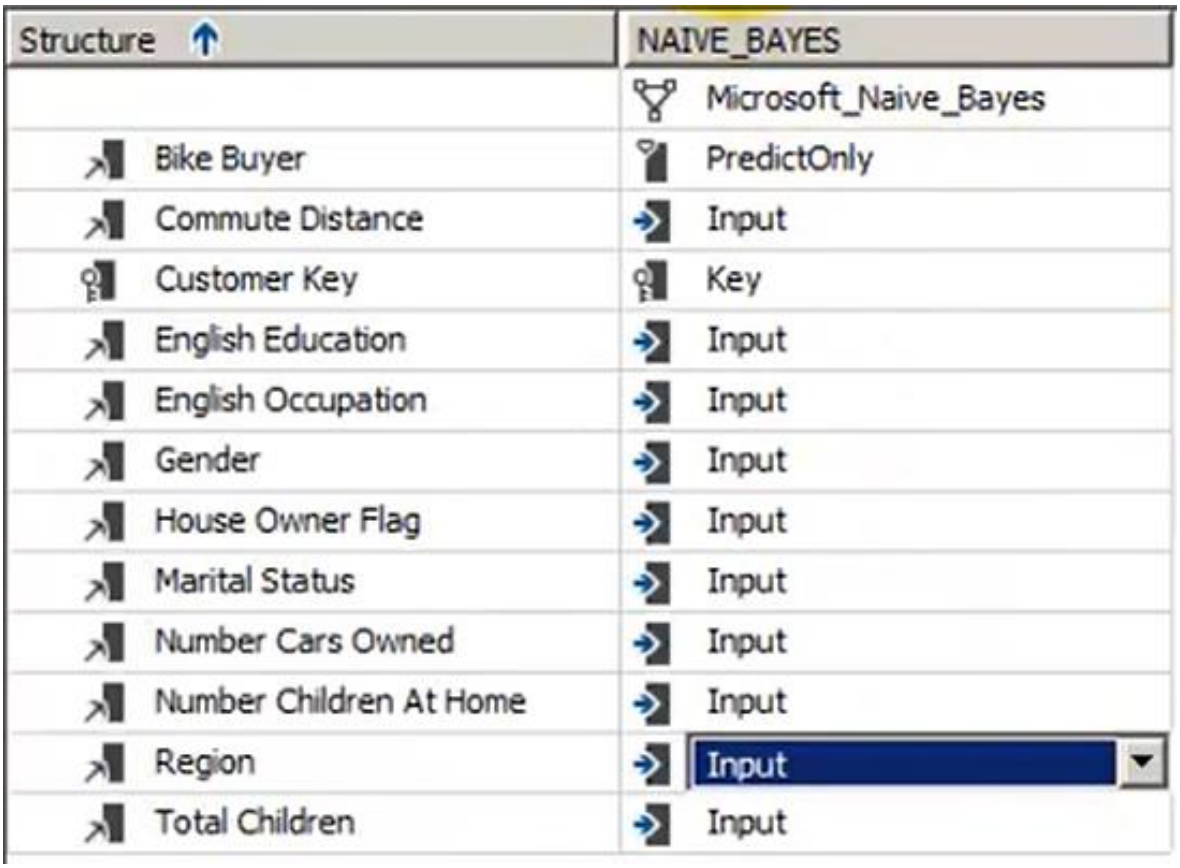


Рисунок 2.31 – Вибір вхідних, ключових та вихідних даних для спрощеного алгоритму Байеса

При застосуванні спрощеного алгоритму Байеса до моделі, можна спостерігати картину, схожу на алгоритм дерева прийняття рішень, на якій показано «найсильніші» та «найслабші» зміні (рис. 2.32).

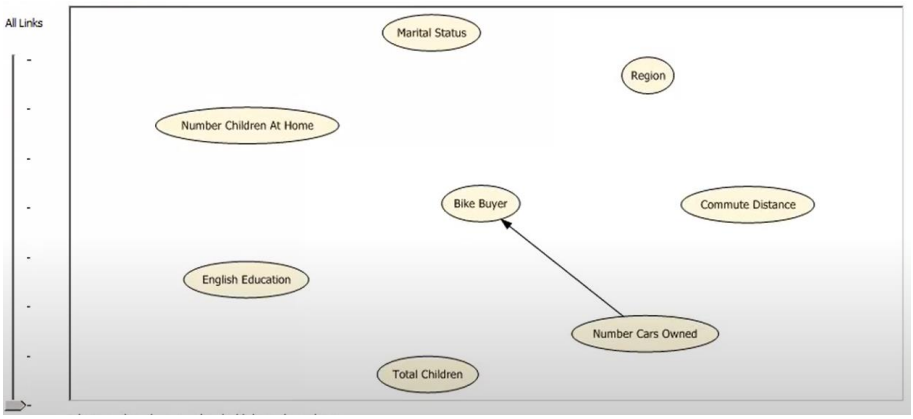


Рисунок 2.32 – «Найсильніша» зміна для спрощеного алгоритму Байеса

Також можна подивитись як кожне значення поля розподіляється в розрізі передбачуваних даних, в даному випадку - прапору покупки велосипеда (рис. 2.33).

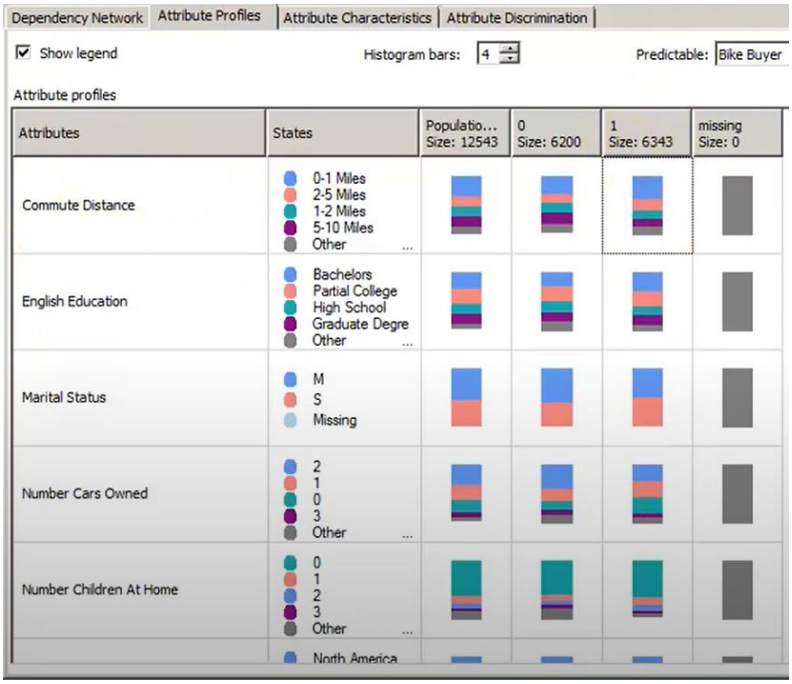


Рисунок 2.33 – Модель з використанням спрощеного алгоритму Байеса

2.3.3 Порівняльний аналіз

Отже, після реалізації та дослідження задачі прогнозування дискретного атрибуту з використанням алгоритму дерева прийняття рішень, алгоритму кластеризації, алгоритму нейронної мережі та спрощеного алгоритму Байеса. До подальшого порівняння будуть входити лише перші три алгоритми з переліку, оскільки з використанням спрощеного алгоритму Байеса можна здебільшого переглянути концентрацію клієнтів в розрізі всіх полів, тому цей алгоритм доцільно використовувати для більш детального аналізу вибірки даних.

З використанням можливості SSAS присвоювати оцінку ефективності використаних алгоритмів (див. рис. 2.30) було протестовано алгоритми дерева прийняття рішень, алгоритм кластеризації та алгоритм нейронної мережі на одній вибірці з різною кількістю застосованих полів для прогнозування дискретного атрибуту. За отриманими результатами можна побудувати наступний графік (рис. 2.34).

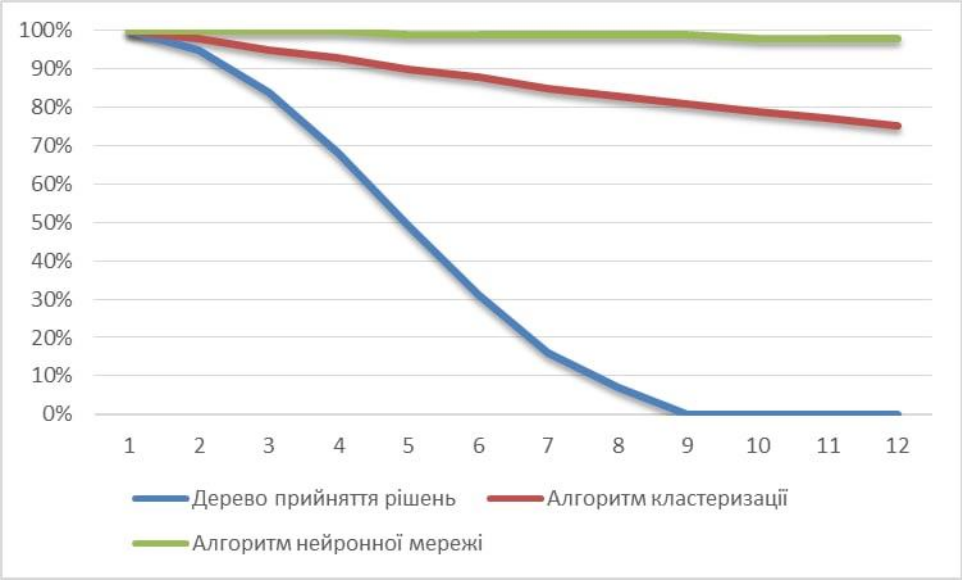


Рисунок 2.34 – Порівняння алгоритмів дерева прийняття рішень, алгоритму кластеризації та алгоритму нейронної мережі

Тому для прогнозування дискретного атрибуту при великої кількості змінних краще використати алгоритм нейронної мережі, а чим менше змінних тим алгоритм дерева прийняття рішень буде кращим.

2.4 Задача знаходження груп схожих елементів в транзакціях

В даному розділі було вирішено задачу використання аналізу кошику споживача для встановлення місць розміщення продуктів та задачу виявлення додаткових продуктів, які можна запропонувати клієнту з використанням алгоритму взаємозв’язків.

2.4.1 Алгоритм взаємозв'язків

Для даного алгоритму обрано дві таблиці (рис. 2.35):

- vAssocSeqLineItems – таблиця з порядком, за яким було придбано товари в чеку;
- vAssocSeqOrders – таблиця з інформацією про чек.

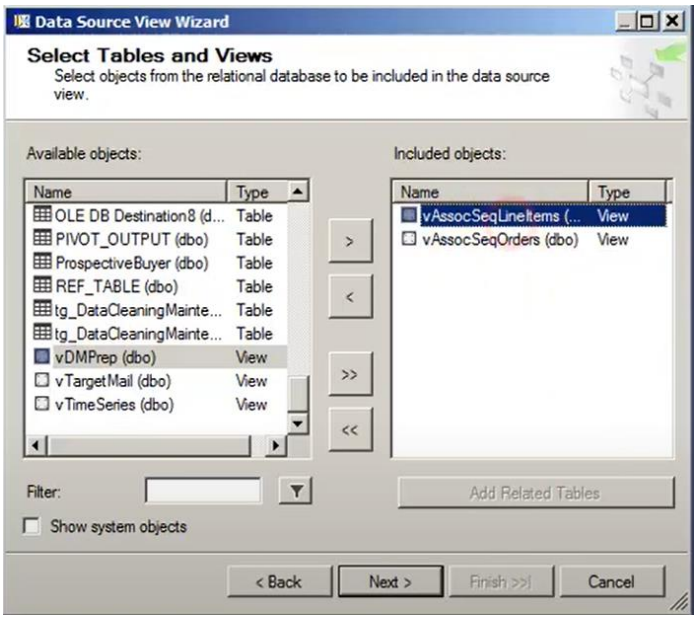


Рисунок 2.35 – Вибір таблиць для алгоритму взаємозв’язків

Та було проведено з’єднання цих таблиць по одному ключу – OrderNumber (рис. 2.36).

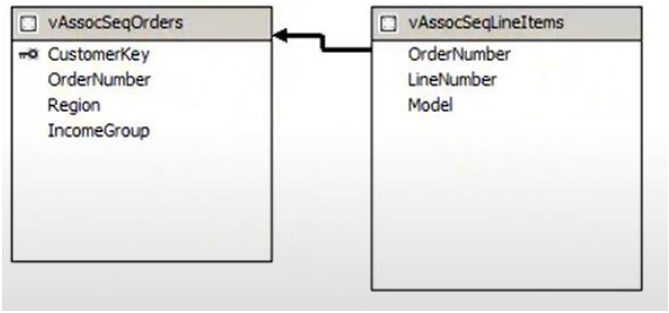


Рисунок 2.36 – З’єднання таблиць за одним ключом

Наступним кроком потрібно обрати тип таблиць (рис. 2.37).

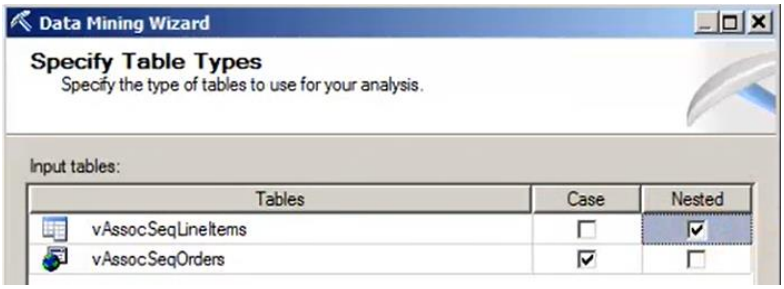


Рисунок 2.37 – Обрання типів таблиць для алгоритму взаємозв’язків

Далі необхідно обрати типи полів (рис. 2.38).

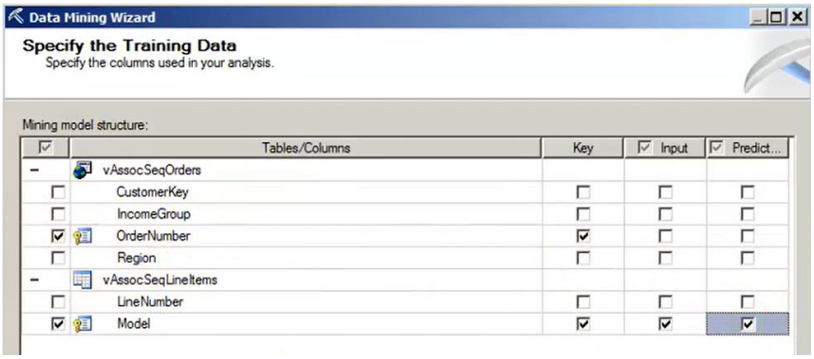


Рисунок 2.38 - Вибір вхідних, ключових та вихідних даних для алгоритму взаємозв’язків

Після всіх налаштувань було отримано наступну структуру для видобутку даних (рис. 2.39).

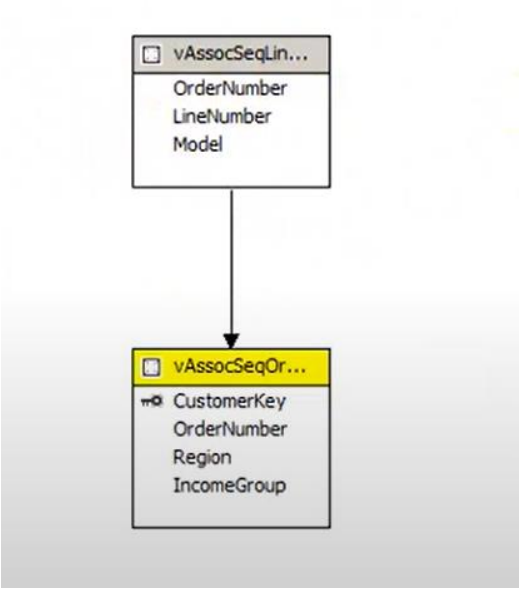


Рисунок 2.39 - Структура для моделі видобутку даних з використанням алгоритму взаємозв’язів

Після розгортання проекту, можна спостерігати наступну аналітику (рис. 2.40), в якій показано набір правил, за яким клієнту варто запропонувати конкретний продукт в залежності від заповненої корзини.

Probability	Importance	Rule
1.000		Road-250, Road Tire Tube -> HL Road Tire
1.000		Touring-1000, Touring Tire Tube -> Touring Tire
1.000		Mountain-500, Mountain Tire Tube -> LL Mountain Tire
1.000		Road-350-W, Road Tire Tube -> ML Road Tire
1.000		Road-750, Road Tire Tube -> LL Road Tire
1.000		Road-550-W, Road Tire Tube -> ML Road Tire
1.000		Touring-3000, Touring Tire Tube -> Touring Tire
1.000		Touring-2000, Touring Tire Tube -> Touring Tire
1.000	1.	Mountain-400-W, Mountain Tire Tube -> ML Mountain Tire
1.000	1.	Mountain-200, Mountain Tire Tube -> HL Mountain Tire
1.000	1...	Touring Tire, Sport-100 -> Touring Tire Tube
1.000	1...	Road-750, Water Bottle -> Road Bottle Cage
1.000	1....	Touring Tire, Half-Finger Gloves -> Touring Tire Tube
1.000	1....	Hydration Pack, Touring Tire -> Touring Tire Tube
1.000	1....	Touring Tire, Cycling Cap -> Touring Tire Tube
1.000	1.183	Mountain-200, Water Bottle -> Mountain Bottle Cage
1.000	1.176	Touring Tire, Road Bottle Cage -> Touring Tire Tube
1.000	1.175	Touring-1000, Water Bottle -> Road Bottle Cage
1.000	1.173	Touring Tire, Water Bottle -> Touring Tire Tube
1.000	1.141	Road-250, Water Bottle -> Road Bottle Cage

Рисунок 2.40 – Модель з використанням алгоритму взаємозв’язків

2.5 Задача прогнозування послідовностей

В даному розділі розглядається вирішення задачі аналізу послідовності придбання товарів в кошику покупця з використанням алгоритму кластеризації послідовностей.

2.5.1 Алгоритм кластеризації послідовностей

В даному алгоритмі буде використовуватись так ж сама вибірка, що і в минулому розділі до кроку з вибором типів полів, що буде виглядати наступним чином (рис. 2.41).

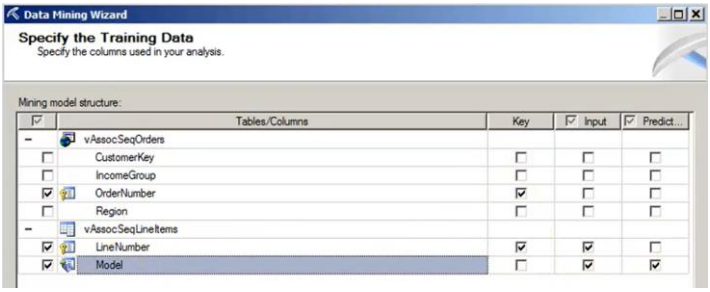


Рисунок 2.41 - Вибір вхідних, ключових та вихідних даних для алгоритму кластеризації послідовностей

Після розгорнення проекту буде побудована наступна модель (рис. 2.42), в якій відбулось розбиття вибірки на кластери.

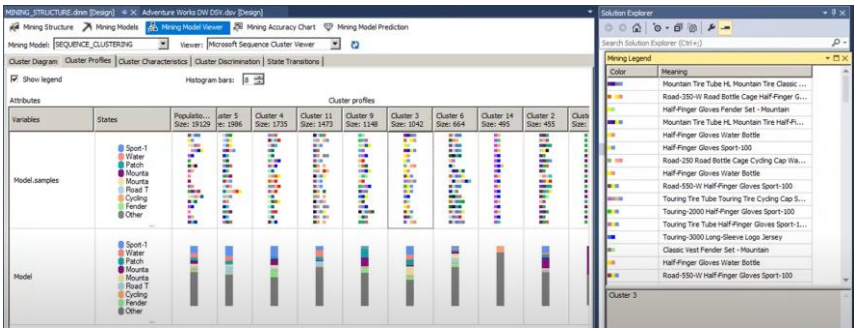


Рисунок 2.42 – Модель з використанням алгоритму кластеризації послідовностей у вигляді кластерів

Також наявна можливість перегляду отриманих даних в числовому вигляді на прикладі третього кластеру (рис. 2.43).

Characteristics for Cluster 3

Variables	Values	Probability
Model.Transitions	[Start] -> Mountain-200	
Model.Transitions	[Start] -> LL Road Tire	
Model	Sport-100	
Model	Mountain-200	
Model.Transitions	[Start] -> Touring-2000	
Model	Patch kit	
Model	Water Bottle	
Model.Transitions	missing	
Model	Road Bottle Cage	
Model.Transitions	[Start] -> Road-350-W	
Model	LL Road Tire	
Model	Fender Set - Mountain	
Model.Transitions	[Start] -> Touring-1000	
Model.Transitions	missing	
Model.Transitions	Road Bottle Cage,Water Bottle	
Model.Transitions	Mountain-200,Patch kit	
Model.Transitions	[Start] -> Mountain Tire Tube	
Model.Transitions	[Start] -> Road-250	
Model	Touring-2000	
Model	Half-Finger Gloves	
Model.Transitions	LL Road Tire,Patch kit	

Рисунок 2.43 – Модель з використанням алгоритму кластеризації послідовностей у вигляді ймовірностей

Найбільш зручним способом візуалізації отриманих даних є наступна діаграма (рис. 2.44), на якій можна обрати продукт і подивитись з якою ймовірністю та який продукт придбають наступним.

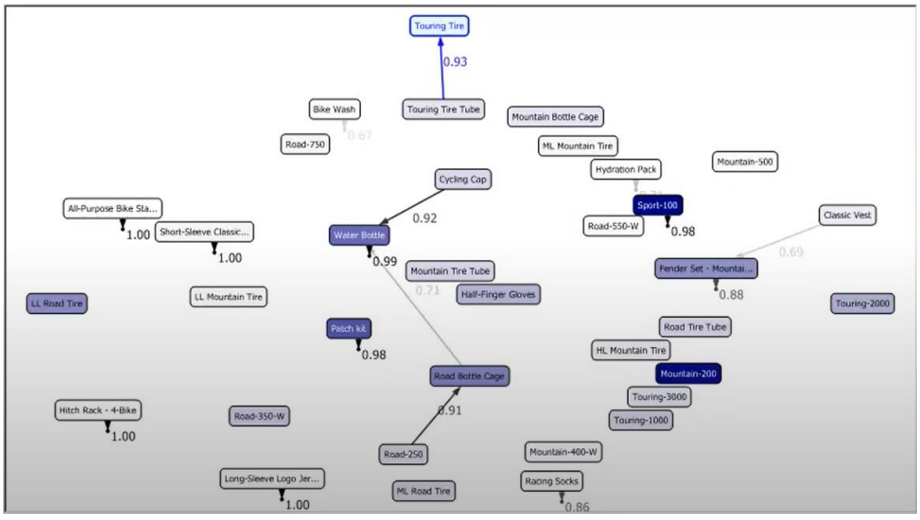


Рисунок 2.44 – Візуалізація моделі з використанням алгоритму кластеризації послідовностей у вигляді діаграми

2.6 Висновки

В результаті використання вказаних алгоритмів – їх реалізація та тестування їхньої роботи на тестових моделях, було отримано наступні результати.

Було протестовано наступні алгоритми:

- Алгоритм кластеризації
- Алгоритм дерева прийняття рішень
- Алгоритм лінійної регресії
- Алгоритм взаємозв'язків
- Спрощений алгоритм Байеса
- Алгоритм нейронної мережі
- Алгоритм кластеризації послідовностей
- Алгоритм часових рядів

На основі досліджених алгоритмів було реалізовано такі типи задач:

- Прогнозування дискретного атрибуту (помітка клієнтів із списку потенційних покупців як хороших і поганих кандидатів)
    - Алгоритм дерева прийняття рішень
    - Алгоритм кластеризації
    - Алгоритм нейронної мережі
    - Спрощений алгоритм Байеса
- Прогнозування безперервного атрибуту (прогноз продаж на наступний рік та формування оцінки ризику з врахуванням демографії)
    - Алгоритм дерева прийняття рішень
    - Алгоритм лінійної регресії
    - Алгоритм часових рядів
- Прогнозування послідовності (аналіз послідовності придбання товарів в кошику клієнта)
    - Алгоритм кластеризації послідовностей

- Знаходження груп спільних елементів в транзакціях (виявлення додаткових продуктів, які можна запропонувати клієнту та застосування аналізу кошику клієнта для встановлення місць розміщення продуктів)
    - Алгоритм взаємозв'язків
    - Алгоритм дерева прийняття рішень

Після реалізації цих задач отримано відповідні висновки для відповідних задач:

- Для прогнозування дискретного атрибуту – при великої кількості змінних краще використати алгоритм нейронної мережі, а чим менше змінних тим алгоритм дерева прийняття рішень буде кращим
- Для прогнозування безперервного атрибуту:
    - Для формування оцінки ризику з врахуванням демографії кращим вибором буде алгоритм дерева прийняття рішень, оскільки точність в обох варіантах однакова, але алгоритм дерева прийняття рішень більш гнучкий у використанні
    - Для прогнозу продаж на наступний рік кращим вибором буде алгоритм часових рядів, оскільки інші алгоритми для даного типу задач не працюють
- Для прогнозування послідовності кращим вибором буде алгоритм кластеризації послідовностей, оскільки для даної задачі інші алгоритми не працюють
- Для знаходження груп спільних елементів в транзакціях – кращим вибором буде алгоритм взаємозв’язків, а не алгоритм дерева прийняття рішень, оскільки він на початку охоплює більшу кількість інформації і тому за умови однакової точності він буде кращим

3 СТАРТАП-ПРОЕКТ

3.1 Ідея проекту

В даному розділі представлене економічне обґрунтування стартапу «Angry score». Поточний розділ написаний з метою ознайомлення з функціональними та економічними характеристиками стартапу, що знаходиться в розробці, аспектами економіки реалізації проекту та підготовки до використання на практиці.

Опис основної ідеї стартапу, що розробляється, показано в Таблиці 3.1.

Таблиця 3.1 - Опис ідеї стартапу

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Система, що допоможе компаніям визначити ступінь довіри до клієнта	Розрахунок найвигідніших та найменш ризикових пропозицій для клієнта	Наявність найвигідніших пропозицій, що може представити дана компанія, що дозволяє зекономити час на пошуки
	Полегшення роботи верифікаторів	Полегшення взаємодії з базою документів

В таблиці міститься опис ідеї стартапу «Angry score». Ціль даної таблиці - ознайомити з функціональними та економічними вигодами стартапу, що знаходиться в розробці.

Дослідження можливих технічно-економічних переваг ідеї (в чому особливість в порівнянні з вже розробленими аналогами) в порівнянні з конкурентними існуючими ідеями полягає в:

- встановленні минулого конкурентного кола або товарів, що можуть замінити власний чи бути його аналогом, проведенні інформаційного збору значень технічно-економічних характеристик для ідеї стартапу, що розробляється, та конкурентних стартапів спираючись на перелік вище;

- встановленні переліку технічно-економічних ознак та характеристик стартапу;
- проведенні порівняльного аналізу характеристик: для власного стартапу встановлюються показники, в яких міститься: а) кращі значення (S, сильні); б) гірші значення (W, слабкі); в) аналогічні (N, нейтральні) (табл. 3.2).

Таблиця 3.2 - Визначення слабких, нейтральних та сильних властивостей ідеї стартапу

№ п/п	Техніко-економічні характер-риситики ідеї	(Потенційні) товари/концепції конкурентів					W слабка сторона	N нейтральна сторона	S сильна сторона
		Мій Проект	К-Т 1	К-Т 2	К-Т 3	К-Т 4			
1.	Форма виконання	Додаток	Веб-Додаток	Додаток	Веб-Сервіс	Веб-Сервіс		+	
2.	Собівартість	Низька	Висока	Висока	Низька	Середня			+
5.	Кросплатформеність	Так	Ні	Ні	Так	Ні			+
6.	Масштабованість	Так	Ні	Ні	Ні	Ні			+
	Кількість змінних для аналізу	велика	Низька	велика	низька	низька	+		

Зазначений аудит слабких, нейтральних та сильних властивостей та характеристик ідеї потенційного стартапу є основою для становлення його конкурентоспроможним.

Слабким місцем для стартапу є велика кількість змінних для вибірки клієнтів, але при збільшенні кількості змінних збільшується точність оцінки ступеню довіри до клієнта.

3.2 Технологічний аудит ідеї проекту

В даному розділі необхідно проаналізувати перелік технології, з використанням якої можна буде реалізувати стартап (додаток для скорингу клієнтів). Було проаналізовано наступні складові: (таблиця 3.3)

- яка технологія буде використана для розробки стартапу?
- чи є потрібні технології. Можливо потрібно доробити/зробити?
- чи є дані технології у відкритому доступі?

Таблиця 3.3 - Технологічна здійсненність ідеї стартапу

№ п/п	Ідея проекту	Технології її реалізації	Наявність технології	Доступність технології
1.	Створення додатку для скорингу клієнтів підприємства	MSSQL, Python	Наявна	Безкоштовна, доступна
		Oracle(Pl/SQL), Java	Наявна	Частково платна, доступна
		Python, Postrges, Django	Наявна	Безкоштовна, доступна
Для створення додатку обрані технології (MSSQL, Python), які є безкоштовними, доступними та добре дослідженими потенційними розробниками з гарною документацією.				

Згідно з аналізом таблиці можна зробити заключення про технологічну реалізацію проекту: технологічного шляху, за яким це доречно виконати (з перечислених технологій обираються наступні: технологія повинна бути у відкритому доступі).

3.3 Аналіз ринкових можливостей

Встановлення можливостей ринку, які можна застосувати в момент впровадження стартапу у відкритий доступ, та загроз на ринку, що мають вірогідність завадити реалізації стартапу, дозволяє спрогнозувати можливі

шляхи розвитку стартапу з бранням до уваги стан ринкового середовища, потреб очікуваних клієнтів та пропозицій стартапів- конкурентів.

В першу чергу потрібно провести аналітику попиту: наявність попиту на ринку, обсяг продажу, динаміка розвитку ринку (табл. 3.4).

Медіанне значення рентабельності порівнюється із відсотком на вкладення банку. якщо перший є нижчим, можливо, є сенс інвестувати в інший стартап.

Після аналізування даних таблиці виноситься висновок про те, чи є ринок сприятливим для входження. Так, ринок є сприятливим для просування даного стартапу.

Таблиця 3.4 - Попередня характеристика потенційного ринку стартапу

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	4
2.	Загальний обсяг продаж, грн/ум.од	20000 грн./ум.од
3.	Динаміка ринку (якісна оцінка)	Зростає/спадає/стагнує
4.	Наявність обмежень для входу (вказати характер обмежень)	Немає
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	R = (3000000 * 100) / (1000000 * 12) = 25%

Потім визначаються цільові клієнтські групи, їх параметри, та визначається приблизний список вимог до товару (табл. 3.5).

Таблиця 3.5 - Параметри потенційних клієнтів стартапу

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Оцінка ступеню довіри до клієнта	Фінансові установи	Не всі захочуть переходити з вже наявних систем	Надійність та конфіденційність процесу

Після того як було визначено потенційні групи клієнтів потрібно провести аналіз середовища ринку: потім скласти таблиці факторів, що допомагають в впровадженню стартапу в середовище ринку, та факторів, що можуть цьому перешкодити (табл. №№ 3.6-3.7).

Таблиця 3.6 - Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Випередження	Вихід на ринок конкурентів першими	1) Передбачити переваги власного ПЗ в порівнянні з конкурентами 2) Передбачити недоліки конкурентів 3) Вийти з ринку
2.	Неможливість використати заплановані технології	Виявлення невідповідності вимогам або зміни в доступності	1) Розглядати аналоги обраних технологій 2) Будувати систему з оглядом на можливість швидкої заміни компонентів у разі необхідності
3.	Держава	Зростання податкового тягаря	Перегляд виконання умов, що зменшують податки
			Поступове підвищення/створення тарифів

Таблиця 3.7 - Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання можливостей потенційних покупців	Зростання держфінансування у галузі розподілених обчислень	Запропонувати свої послуги державним підприємствам
2.	Покращення технології пошуку	Пришвидшення відгуку системи	Розробити рекламну кампанію враховуючи свою перевагу
3.	Держава	Послаблення обмежень в законодавстві	Оптимізація діяльності для скорочення витрат

Потім аналізуються пропозиції: йде визначення загальних рис конкуренції (таб. 3.8)

Таблиця 3.8 - Ступеневий аналіз конкуренції на ринку

Особливість конкурентного середовища	В чому проявляється характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – чиста	Існує 4 конкуренти на ринку	Врахувати ціни конкурентних компаній на початкових етапах створення бізнесу, реклама (вказати на конкретні переваги перед конкурентами)
2. За рівнем конкурентної боротьби - міжнародний	Всі компанії з інших країн	Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях
3. За галузевою ознакою – внутрішньогалузева	Конкуренти мають ПЗ, яке використовується лише всередині даної галузі	Створити основу ПЗ таким чином, щоб можна було легко переробити дане ПЗ для використання у інших галузях
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Види товарів є однаковими, а саме – програмне забезпечення	Створити ПЗ, враховуючи недоліки конкурентів
5. За характером конкурентних переваг – неціновa	Вдосконалення технології створення ПЗ	Використання менш дорогих технологій для розробки в порівнянні з конкурентами
6. За інтенсивністю – марочна	Бренди присутні	-

Після проведення аналізу конкурентності проводиться детальне аналізування умов конкуренції (за моделлю 5 сил М. Портера) (табл. 3.9).

Таблиця 3.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти (бар'єри входження в ринок)	Фактори сили постачальників	Фактори сили споживачів	Фактори загроз з боку товарів-замінників
Висновки	Існує 4 конкуренти на ринку. Найбільш схожим за виконанням є конкурент 1, так як його рішення також має високу швидкодію	Так, можливості для входу на ринок є, бо розроблене рішення дозволяє більш оптимально використовувати наявні обчислювальні ресурси.	Постачальники відсутні	Важливим для користувача є швидкість та коректність роботи програми	Товари замінники можуть використати більш дешеву технологію створення ПЗ та зменшити собівартість товару або вийти першими на ринок задовольнивши клієнтів

Відповідно до проаналізованих даних таблиці виводиться висновок про можливості праці на ринку зважаючи на конкурентів. Також висновок про характеристики (плюси), які має містити проект, щоб мати змогу конкурувати з іншими на ринку. Другий висновок береться до уваги при визначенні переліку факторів конкурентності.

На базі аналізу конкуренції (табл. 3.9), а також беручи до уваги характеристики ідеї стартапу (табл. 3.2), вимог споживачів (табл. 3.5) та факторів середовища маркетингу (табл. №№ 3.6-3.7) отримується перелік факторів конкурентності показано в табл. 4.10.

Таблиця 4.10 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Кросплатформність	СУБД MSSQL
2.	Масштабованість та адаптивність	Без додаткових зусиль розширюється як функціональність проекту, так і апаратне забезпечення, необхідне для роботи

За отриманими факторами конкурентності (табл. 3.10) аналізуються плюси та мінуси стартапу (табл. 3.11), за застосування оцінки конкурентоспроможності за 20-бальною шкалою.

Таблиця 3.11 - Порівняльний аналіз сильних та слабких сторін стартапу «Angry score»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з нашим підприємством						
			-3	-2	-1	0	+1	+2	+3
1.	Кросплатформність	20			+				
2.	Масштабованість та адаптивність	15		+					

Кінцевим етапом аналізування можливостей впровадження стартапу є проведення SWOT-аналізу (аналіз сильних (Strength), слабких (Weak) сторін, можливостей (Opportunities) та загроз (Troubles) (табл. 3.12) на базі виділених загроз ринку, плюсів та мінусів (табл. 3.11). Аудит загроз ринку та можливостей ринку складається на базі проаналізованих факторів загроз. Загрози ринку та можливості ринку є результатами (прогнозованими наслідками) впливу чинників, і, ще не є розгорнутими на ринку та мають велику вірогідність успішного завершення.

Таблиця 3.12 – SWOT-аналіз стартап-проекту

S	Наявність декількох моделей для скорингу, у користувача є можливість обрати потрібну саме йому	Для точної оцінки клієнта потрібна велика кількість змінних	W
O	Таргетингова реклама, акцентування уваги на надійності системи, заохочення співробітників конкурентів до зміни компанії	Загрози: конкуренція, зміна потреб користувачів	T

На базі SWOT-аналізу виконуються аналоги ринкової поведінки для впровадження стартапу на ринок та приблизний час для їх реалізації дивлячись на проекти конкурентів,які можуть бути впроваджені на ринок (див. табл. 4.9). отримані аналоги аналізуються з точки зору часу та вірогідності отримання ресурсів (табл. 3.13).

Таблиця 3.13 – Аналоги ринкового впровадження стартапу

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення проекту використовуючи іншу СУБД	80%	8 місяців
2.	Створення проекту використовуючи NoSQL	55%	10 місяців

З визначених аналогів вибирається така, для якої: а) більш ймовірним та простим отримання ресурсів ; б) більш стислі строки реалізації. Обираємо перший аналог.

3.4 Розробка ринкової стратегії проекту

Робота над стратегією ринка початковим кроком передбачає встановлення стратегії ринкового охоплення: опис планових груп можливих користувачів (табл. 3.14).

Таблиця 3.14 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Банки	Спрощення процесу встановлення степені довіри до клієнта	Великий	Існує 4 конкуренти, які надають схожі, але менш швидкі та якісні рішення.	Може бути непросто через велику кількість конкурентів
2.	Мікрофінансові установи	Спрощення процесу відмови у видачі кредиту клієнтам	Середній		
Обрано цільові групи: Банки					

Згідно з аналізом потенційних груп споживачів (сегментів) розробники та маркетологи визначать цільові групи, котрим вони запропонують свій товар, та встановлюють стратегію для охоплення ринку:

- коли компанія зосереджена на конкретному сегменті – вона вибирає стратегію концентрованого маркетингу;
- коли компанія веде діяльність із кількома сегментами і веде розробку для них окремо – то компанія обирає стратегію диференційованого маркетингу;
- коли компанія веде діяльність з усім ринком, – вона застосовує масовий маркетинг.

Таблиця 3.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Створення додатку з використанням методів Data mining	Ринкове позиціонування	Швидкість, якість, розподіленість	Диференціація

Для функціонування в обраних сегментах потрібно створити базову стратегію розвитку (табл. 3.15).

Наступним кроком є обґрунтований вибір стратегії конкурентної поведінки (табл. 3.16)

Таблиця 3.16 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні	Так	Буде, а саме високу швидкість роботи конкурента 1	Зайняття конкурентної ніші

На базі вимог споживачів до постачальника (компанії - стартапу) і також до продукту (див. табл. 3.5), залежно від вибраної опорної стратегії конкурентної поведінки (табл. 3.16) та стратегії розвитку (табл. 3.15) та готується стратегія позиціонування (табл. 3.17). що несе у собі формування позиції ринку, за якими люди повинні розпізнати торгівельну марку/стартап.

Таблиця 3.17 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартапу	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	Швидкість	Диференціація	Система дозволяє отримувати результат в реальному часі, навіть у випадку проблем з комунікацією	Швидкість, розподіленість

Результатом виконання має вийти погоджена система рішень про ринкову поведінку компанії-стартапу, яка буде визначати напрямки функціонування компанії-стартапу на ринку.

3.5 Розробка маркетингової програми

На старті потрібно сформувати маркетингову концепцію товару, який у майбутньому повинен отримати користувач. Саме тому у Таблиці 4.18 важливою умовою є підбивання підсумків попереднього аналізування конкурентоспроможності товару.

Таблиця 3.18 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Швидкість обробки даних	ПЗ дозволяє працювати в режимі реального часу	Переваги у швидкості
2.	Можливість окремої роботи кластерів	Кожен кластер може деякий час працювати відокремлено	Переваги у надійності

Потім визначається трирівнева маркетингова модель товару: узагальнюється ідея послуги та/або продукту, фізичні складові товару, процес його придбання (табл. 3.19), приблизний перелік ймовірних характеристик товару.

Таблиця 3.19 - Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
І. Товар за задумом	Система дозволяє більш оптимально використовувати обчислювальні ресурси та не потребує постійної синхронізації для правильної роботи та стабільно працювати у випадку втрати зв'язку між кластерами

Таблиця 3.19 (закінчення)

1	2		
II. Товар у реальному виконанні	Властивості / характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Швидкість	Нм	Технологічна
	2. Розподіленість	Нм	Технологічна
	3. Надійність	Нм	Технологічна
	4. Простота	Нм	Технологічна
	Якість: згідно до стандарту ISO 4444 буде проведено тестування		
	Пакування відсутнє		
	Моя компанія: “Angry score”		
III. Товар із підкріплюванням	1-місячна пробна безкоштовна версія та безкоштовне встановлення, безкоштовні ліцензія для банків		
	Постійна підтримка для користувачів		
За рахунок чого потенційний товар буде захищено від копіювання			

Після розробки маркетингової моделі товару потрібно відмітити – яким чином проект буде захищено від недозволеного копіювання. Захист можна організовувати як захист ідеї товару (захист інтелектуальної власності), чи ноу-хау, чи поєднання характеристик і властивостей.

Потім потрібно визначення цінові межі, які потрібно враховувати при визначенні ціни на товар (фінальний розрахунок ціни відбудеться під час фінансово-економічного аналізу проекту), котре має за мету аналізування ціни на товари-аналоги або товари замінники, також проводиться аналіз доходів цільової групи (табл. 3.20). Аналіз проводиться з застосуванням експертного методу.

Таблиця 3.20 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники, грн	Рівень цін на товари-аналоги, грн	Рівень доходів цільової споживачів, грн групи	Верхня та нижня межі встановлення ціни на товар/послугу, грн
1.	25000	30000	200000	20000

Потім потрібно визначити найкращу систему збуту (табл. 3.21):

- збувати товар власними силами або з використанням посередників (власна або залучена система збуту);
- обґрунтування вибраної найкращої глибини каналу збуту;
- обґрунтування вибраних посередників.

Таблиця 3.21 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Купують ПЗ та роблять внески в залежності від наданої вибірки клієнтів	Продаж	0 (напряму) 1 (через одного посередника)	Власна та через посередників

Фінальною складовою є розробка концепції маркетингових комунікацій, що базується на попередній основі для позиціонування, зазначену специфіку поведінки користувачів (табл. 3.22).

Таблиця 3.22 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Купівля ПЗ через інтернет	Інтернет	Швидкість Розподіленість Надійність Простота	Показати переваги програми, у тому числі і в порівнянні з конкурентами	Деморолик із використанням

3.6 Висновки

В межах даного дослідження були знайдені основні фінансові показники проекту, також було оглянути ризики які можуть нести загрозу для проекту. Згідно з дослідженнями:

- проект має потенціал та багато шансів для комерційного успіху на ринку
- проект є рентабельним в умовах ринку
- проект є конкурентним, бар’єри входження низькі
- є вагомі перспективи для реалізації на ринку з огляду на потенційних клієнтів
- подальша імплементація є доцільною
- щоб проект був успішно виконаний, потрібно реалізувати програму використавши додаткові бібліотеки для SQL

ВИСНОВКИ

Основним завданням даної роботи було дослідження та використання алгоритмів Data mining, які надаються сервісом SSAS та з використанням Microsoft Visual Studio. В ході дослідження задача була розкрита повною мірою на прикладі реалізації алгоритмів Data mining для конкретних прикладних задач.

Були розглянуті різні алгоритми Data mining та обрані найбільш актуальні та найбільш вживані в нинішніх реаліях.

Практична частина дала змогу зрозуміти для яких саме задач які алгоритми краще підходять:

- Для прогнозування дискретного атрибуту – при великої кількості змінних краще використати алгоритм нейронної мережі, а чим менше змінних тим алгоритм дерева прийняття рішень буде кращим
- Для прогнозування безперервного атрибуту:
    - Для формування оцінки ризику з врахуванням демографії кращим вибором буде алгоритм дерева прийняття рішень, оскільки точність в обох варіантах однакова, але алгоритм дерева прийняття рішень більш гнучкий у використанні
    - Для прогнозу продаж на наступний рік кращим вибором буде алгоритм часових рядів, оскільки інші алгоритми для даного типу задач не працюють
- Для прогнозування послідовності кращим вибором буде алгоритм кластеризації послідовностей, оскільки для даної задачі інші алгоритми не працюють
- Для знаходження груп спільних елементів в транзакціях – кращим вибором буде алгоритм взаємозв’язків, а не алгоритм дерева прийняття рішень, оскільки він на початку охоплює більшу кількість інформації і тому за умови однакової точності він буде кращим

Виходячи з результатів тестування, звичайно, можна сказати, що якийсь алгоритм є кращим іншого, але це не буде правдиво на 100%, оскільки в деяких випадках з певними умовами інші алгоритми могли показатися кращими.

ПЕРЕЛІК ПОСИЛАНЬ

1. Agrawal R., Imielinski T., Swami A. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering. 1993. 5, No 6. P. 914–925. Special Issue on Learning and Discovery in Knowledge Based Databases.
2. Agrawal R., Imielinski T., Swami A. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C., May 1993.
3. Agrawal R., Srikant R. Fast algorithms for mining association rules in large databases. Research Report RJ 9839, IBM Almaden Research Center, San Jose, California, June 1994.
4. Agrawal R., Srikant R. Mining Sequential Patterns. Proc. of the Int'l Conference on Data Engineering (ICDE). Taipei, Taiwan. 1995.
5. Analysis of Data Mining Algorithms : веб-сторінка. URL: https://www-users.cs.umn.edu/~desi0016/research/dataminingoverview.html (дата звернення: 20.09.2020)
6. Bakshi K. A List of top Data Mining Algorithms : веб-сторінка. URL: https://www.techleer.com/articles/438-a-list-of-top-data-mining-algorithms/ (дата звернення: 25.09.2020).
7. Chaudhuri S. Data Mining and Database Systems: Where is the Intersection?. Bull. Technic. Comm. Data Engineering. 1998. 21, No. 1. P. 4–8.
8. Chaudhuri S., Dayal U. An Overview of Datawarehousing and OLAP Technology. In Sigmod Record. 1997.
9. Netz A., Chaudhuri S., Fayyad U., Bernhard J. Integrating data mining with SQL databases: OLE DB for data mining. International Conference on Data Engineering: веб-сторінка. URL: https://doi.org/10.1109/ICDE.2001.914850 (дата звернення: 10.10.2020).

10. Curotto C. L., Ebecken N. F. F. Implementing Data Mining Algorithms in Microsoft SQL Server. Series: Advances in Management Information. Wit Presss. 2005. Vol. 3. 143 p.
11. Data mining : Wikipedia. URL: https://ru.wikipedia.org/wiki/Data_mining (дата звернення: 08.09.2020).
12. Data Mining Tutorial – Introduction to Data Mining (Complete Guide) : веб-сторінка. URL: https://data-flair.training/blogs/data-mining-tutorial/ (дата звернення: 10.09.2020)
13. Fayyad U. et. al. Advances in Knowledge Discovery and Data Mining, MIT Press. 1996.
14. Gray et al. Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub Totals. In Data Mining and Knowledge Discovery. 1997. 1, No 1. P. 29–53.
15. Han J. Towards On-Line Analytical Mining in Large Databases, to appear.
16. Han J., Cai Y., Cercone N. Knowledge discovery in databases: An attribute oriented approach. In Proc. of the VLDB Conference. Vancouver, British Columbia, Canada. 1992. P. 547– 559,
17. Han J., Kamber M. Data Mining Concepts and Techniques. San Francisco, USA: Morgan Kaufmann Publishers. 2001, ISBN 1558604898.
18. Khurana K., Sharma S. A comparative analysis of association rules Mining Algorithms. International J. Sci. Research Public. 2013. 3, No 5. ISSN 2250-3153.
19. Kohavi R., Sommerfield D., Dougherty J. Data Mining using MLC++ : A machine learning library in C++. Tools with AI. 1996 : веб-сторінка. URL: https://doi.org/10.1109/TAI.1996.560457 (дата звернення: 16.11.2020).
20. MacLennan J., Tang Z. H., Crivat B. Data Mining with Microsoft SQL Server 2008. 2009. 676 p.

21. Mannila H., Toivonen H., Verkamo A. I. Efficient algorithms for discovering association rules. In KDD-94: AAAI Workshop on Knowledge Discovery in Databases. 1994.
22. Meo R., Giuseppe P., Ceri S. A new SQL-like Operator for Mining Association Rules. In Proc. of VLDB96. Mumbai, India. P. 122–133,
23. Microsoft Analysis Services : Wikipedia. URL: https://ru.wikipedia.org/wiki/Microsoft_Analysis_Services (дата звернення: 23.09.2020).
24. Sarawagi S., Thomas S., Agrawal R. Integrating Mining with Relational Database Systems: Alternatives and Implications. In Proc. of ACM Sigmod 98, To appear.
25. Spec I. All About SQL Server Analysis Services (SSAS). In An Easy Manner : веб-сторінка. URL: https://www.spec-india.com/blog/sql-server-analysis-services-ssas (дата звернення: 01.09.2020).
26. Srikant R., Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements. Proc. of the Fifth Int'l Conference on Extending Database Technology (EDBT). Avignon, France. 1996.
27. Using SSAS (SQL Server Analysis Services) Data Mining to Automate Marketing Analysis : веб-сайт. URL: https://www.erpsoftwareblog.com/2014/06/using-ssas-sql-server-analysis-services-data-mining-to-automate-marketing-analysis/ (дата звернення: 02.10.2020).
28. Witten I. H., Frank E., Hall M. A. Data Mining: Practical Machine Learning Tools and Techniques. 3rd edition. Morgan Kaufmann. 2011. p. 664. ISBN 9780123748560.
29. Дюк В., Самойленко А. Data Mining: учебный курс (+CD). СПб.: Изд. Питер. 2001. 368 с.

30. Журавлёв Ю. И., Рязанов В. В., Сенько О. В. Распознавание. Математические методы. Программная система. Практические применения. Москва: Изд. «Фазис». 2006. 176 с. ISBN 5-7036-0108-8.
31. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям (+ CD). СПб.: Изд. Питер. 2009. 624 с.
32. Ситник В. Ф., Краснюк М. Т. Інтелектуальний аналіз даних (дейтамайнінг): Навч. посібник. Київ: КНЕУ. 2007. 376 с.
33. Чубукова И. А. Data Mining: учебное пособие. Москва: Интернет-университет информационных технологий: БИНОМ: Лаборатория знаний. 2006. 382 с. ISBN 5-9556-0064-7.