НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Кафедра автоматизованих систем обробки інформації і управління

(повна назва кафедри, циклової комісії)

**КУРСОВА РОБОТА**

з Оброблення надвеликих масивів даних

(назва дисципліни)

на тему: Аналіз та передбачення вартості автомобілей   
на основі заданих параметрів.

Студента (ки) 5 курсу ІП02-мп групи

напряму підготовки Інженерія програмного забезпечення інформаційно-управляючих систем

спеціальності 121

Ковинєва К.О.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Керівник \_ст.вик. Олійник Ю. О.\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_Оцінка: ECTS\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Члени комісії |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |
|  |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |

Київ - 2020 рік

Національний технічний університет України “КПІ імені Ігоря Сікорського”

(назва вищого навчального закладу)

Кафедра автоматизованих систем обробки інформації і управління

Дисципліна «Оброблення надвеликих масивів даних»

Спеціальність 121 "Інженерія програмного забезпечення"

Курс 5 Група ІП-02мп Семестр 9

**ЗАВДАННЯ**

**на курсову роботу студента**

|  |
| --- |
| Ковинєва Кирила Олексійовича |

(прізвище, ім’я, по батькові)

|  |  |
| --- | --- |
| 1. Тема роботи | Аналіз та передбачення вартості автомобілей на основі заданих параметрів. |
|  | |
|  | |

|  |  |
| --- | --- |
| 2. Строк здачі студентом закінченої роботи | 24.12.2020 |

|  |  |
| --- | --- |
| 3. Вихідні дані до роботи | Датасет: Used Cars Dataset |
|  | |
|  | |
|  | |

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
| Постановка задачі прогнозування цін, розробка процесу обробки даних, |
| пошук даних вартості автомобілей, попередня обробка даних, проектування та створення БД, |
| опис методів аналізу даних, тренування моделі за допомого методу лінійної регрсії, |
| тренування моделі за допомогою методів випадкового лісу та дерева з посиленим градієнтом. |

5. Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

|  |
| --- |
| немає |
|  |
|  |
|  |

|  |  |
| --- | --- |
| 6. Дата видачі завдання | 12.10.2020 |

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Назва етапів курсової роботи | Термін виконання етапів роботи | Підписи керівника, студента |
| 1. | Отримання теми курсової роботи | 12.10.2020 |  |
| 2. | Визначення основних задач курсової роботи | 13.10.2020 -  19.10.2020 |  |
| 3. | Пошук та вивчення літератури з питань курсової роботи | 20.10.2020- 26.10.2020 |  |
| 4. | Розробка процесу обробки даних | 27.10.2020 -  09.11.2020 |  |
| 5. | Розробка моделі БД | 10.11.2020 -  16.11.2020 |  |
| 6. | Розробка ETL процесів | 17.11.2020 - 30.11.2020 |  |
| 7. | Опис методів обробки даних | 01.12.2020 -  07.12.2020 |  |
| 8. | Дослідження ефективності методів обробки даних | 08.12.2020 -  14.12.2020 |  |
| 9. | Підготовка пояснювальної записки | 14.12.2020 - 23.12.2020 |  |
| 10. | Здача курсової роботи на перевірку | 24.12.2020 |  |
| 11. | Захист курсової роботи | 29.12.2020 |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Ковинєв К.О. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

|  |  |  |  |
| --- | --- | --- | --- |
| Керівник |  |  | Олійник Ю. О. |
|  | (підпис) |  | (прізвище, ім’я, по батькові) |

"\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 р.

# **АНОТАЦІЯ**

Пояснювальна записка до курсової роботи: 29 сторінок, 26 рисунків, 12 таблиць, 7 посилань.

Об’єкт дослідження: датасет на основі технології Spark

Мета роботи: закріплення знань у дисципліні «Обробка надвеликих масивів даних» та їх використання у аналізі даних за допомогою технології Spark у вирішенні завдання прогнозування вартості автомобілей, яке включає у себе пошук та обробку даних, проектування та створення БД, аналіз методів вирішення задач регресії, дослідження методів вирішення задач регресії, висунення висновків.

Ключові слова: аналіз даних, машинне навчання, spark, pyspark, лінійна регресія, random forest, gradient-boosted tree

**Зміст**

[**АНОТАЦІЯ** 4](#_Toc60746077)

[**ВСТУП** 6](#_Toc60746078)

[**1. ОСНОВНА ЧАСТИНА** 7](#_Toc60746079)

[**1.1 Постановка задачі** 7](#_Toc60746080)

[**1.2 Розробка процесу обробки даних** 8](#_Toc60746081)

[**1.3 Розробка ETL процесів** 9](#_Toc60746082)

[**1.4** **Розробка структури БД** 14](#_Toc60746083)

[**1.5 Опис методів аналізу даних** 17](#_Toc60746084)

[**1.6 Дослідження ефективності методів аналізу даних** 18](#_Toc60746085)

[**ВИСНОВОК** 24](#_Toc60746086)

[**ПЕРЕЛІК ПОСИЛАНЬ** 25](#_Toc60746087)

[**ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО ПРОДУКТУ** 26](#_Toc60746088)

# **ВСТУП**

Автомобілі є невід’ємною частиною сучасного життя кожної людини. Завдяки ним вирішуються питання транспортування товарів, ресурсів та людей. Тому придбання власного автомобіля для бізнесу або особистого використання є важливою частиною у житті майбутнього власника. Проте, якщо на нові машини ціна встановлюється автомобільними компаніями, то на машини, які є колишніми у використанні, ціна встановлюється колишнім власником.

Ціллю роботи є створення програмного забезпечення для прогнозування ціни на автомобіль на основі його параметрів з використанням методів машинного навчання. Це вирішить проблему власника авто, коли важно встановити приблизну вартість на товар, та проблему покупця на відповідність ціни характеристикам авто.

Для аналізу було використано датасет «Used Cars Dataset» отриманий на ресурсі Kaggle, який містить інформацію про автомобілі, що продавалися у США.

Для розв’язання задачі була застосована технологія Spark, мова програмування Python та MS SQL Server для обробки даних.

# **1. ОСНОВНА ЧАСТИНА**

## **1.1 Постановка задачі**

Задачу виконання розробки програми для прогнозування ціни на автомобіль можна розбити на наступні етапи:

1) Підготовка даних до аналізу: завантаження та попередня обробка даних, що включає у себе видалення параметрів, що не впливають на ціну або мають велику кількість незаповнених строк, видалення строк з відсутньою інформацією категоріальних параметрів, заміна значень категоріальних параметрів на числа, пошук аномалій.

2) Аналіз даних: вибір алгоритмів для вирішення задачі регресії, аналіз даних за допомогою обраних алгоритмів, порівняння якості отриманих результатів, винесення висновків.

## **1.2 Розробка процесу обробки даних**

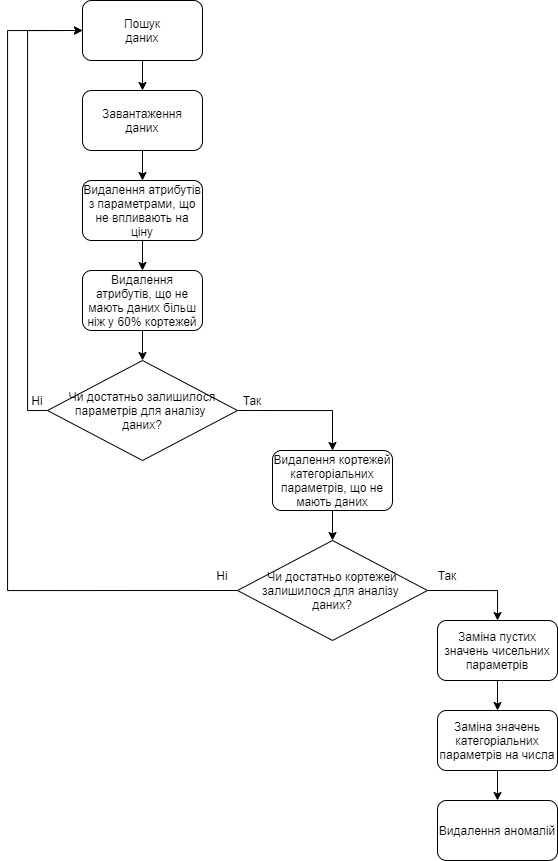


Рис. 1. Процес обробки даних.

## **1.3 Розробка ETL процесів**

**1.3.1 Завантаження даних:**

Для аналізу було обрано датасет «Used Cars Dataset» з електронного ресурсу Kaggle, який містить у собі 24 параметри:

1. id – унікальний номер автомобіля
2. url – URL переліку
3. region – місто продажу авто
4. region\_url – URL міста
5. price – ціна авто
6. year – рік виготовлення авто
7. manufacturer – виробник авто
8. model – модель авто
9. condition – стан авто
10. fuel – тип палива
11. odometer – кількість миль, що авто проїхало
12. title\_status – статус авто
13. transmission – тип трансмісії
14. VIN – ідентифікаційний номер авто
15. drive – тип приводу
16. size – розмір авто
17. type – загальний тип авто
18. paint\_color – кольор авто
19. image\_url – URL зображення авто
20. description – опис авто
21. state – штат продажу
22. let – широта лиістнгу
23. long – довгота лістингу
24. posting date – дата розміщення авто до продажу

**1.3.2 Обробка даних:**

Для обробки даних було використано програму Microsoft SQL Management Studio та мову запитів Transact-SQL.

**Пошук та видалення атрибутів за параметрами, що не впливають на ціну, та параметрів, що не мають даних більш ніж у 60% кортежей:**

1. id – не впливає на ціну, проте потрібен для формування БД.
2. url – не впливає на ціну.
3. region
4. region\_url – не впливає на ціну.
5. price
6. year
7. manufacturer
8. model
9. condition
10. fuel
11. odometer
12. title\_status
13. transmission
14. VIN – не впливає на ціну.
15. drive
16. size – має недостатньо даних (70%)
17. type
18. paint\_color – не впливає на ціну.
19. image\_url – не впливає на ціну.
20. description – не впливає на ціну.
21. state
22. let – не впливає на ціну.
23. long – не впливає на ціну.
24. posting date – не впливає на ціну.

Запит для видалення атрибутів:  
ALTER TABLE dbo.vehicles DROP COLUMN id\_v, url, region\_url, VIN, size,

paint\_color, image\_url, description, lat, long, posting\_date

**Видалення кортежей категоріальних параметрів, що не мають даних:**

Початкова кількість строк даних: 458213  
SELECT COUNT(\*) FROM dbo.vehicles

Запити для видалення строк, що не мають даних:  
DELETE dbo.vehicles

WHERE (manufacturer IS NULL) OR (model IS NULL )

OR (region IS NULL) OR (fuel IS NULL) OR (odometer IS NULL)

OR (state IS NULL) OR (title\_status IS NULL) OR (transmission IS NULL)

OR (condition IS NULL) OR (type IS NULL) OR (drive IS NULL)

**Заміна пустих значень чисельних параметрів:**

Загальним прикладом заміни пустих значень чисельних параметрів є присвоєння середнього значення параметру усього датасету.

Середнім значенням року створення є 2010. Запит для зміни року:  
UPDATE dbo.vehicles

SET year=2010

WHERE year = 0

Так як, вартість авто є параметром, що система повинна навчитися спрогнозувати, задати середнє значення буде невірним рішенням. Запит для видалення строк, де немає ціни:  
DELETE dbo.vehicles

WHERE price IS NULL

**Заміна значень категоріальних параметрів на числа:**

Скористаємося «розумним» способом кодування категорій. Усі категорії будуть кодуватися значенням, що інтерпретуються. У випадку даного датасету значення категорій будуть закодовані середньою вартістю автомобілей. Для зберігання значень створимо базу даних, де кожен категоріальний параметр буде винесений у окрему табличку, що буде зберігати назву категорії та значення середньої вартості для категорії.

**Видалення аномалій:**

Для пошуку аномалій скористаємося графіком залежностей вартості авто від кожного параметру.

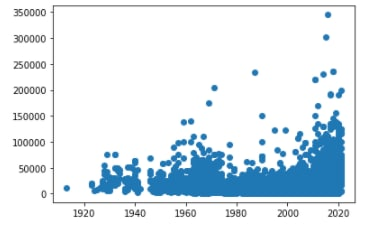


Рис. 2. Графік залежності вартості від року випуску авто.

Є поодинокі точки, що не відносяться до загальної групи, та можуть вплинути на аналіз.

Запити для видалення аномальних точок:

DELETE dbo.vehicles

WHERE (price > 1000000) OR (price < 1000)

DELETE dbo.vehicles

WHERE (price > 125000 AND year < 2000)

DELETE dbo.vehicles

WHERE (price > 150000 AND year > 2000)

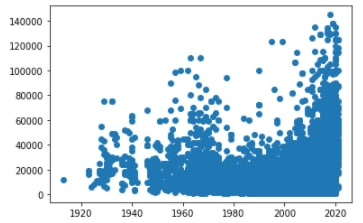


Рис. 3. Графік залежності вартості від року випуску авто після видалення аномальних точок.

Наступними параметрами, де були знайдені аномальні точки, є кількість миль, що проїхало авто, та тип авто.

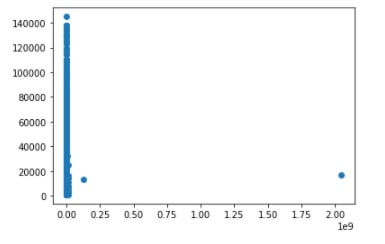


Рис. 4. Графік залежності вартості від кількості миль, що проїхало авто.

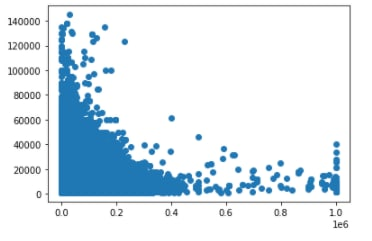


Рис. 5. Графік залежності вартості від кількості миль, що проїхало авто після видалення аномальних точок.

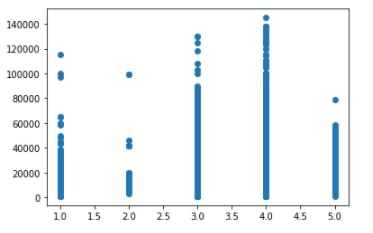


Рис. 6. Графік залежності вартості від типу авто.

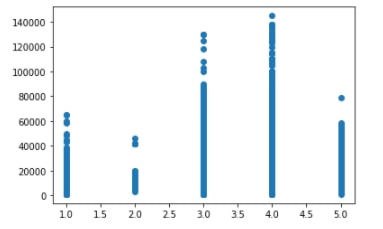


Рис. 7. Графік залежності вартості від типу авто після видалення аномальних точок.

## **Розробка структури БД**

Для використання «розумного» кодування була спроектована та розроблена база даних, де кожна категорія автомобіля була винесена до окремих таблиць:

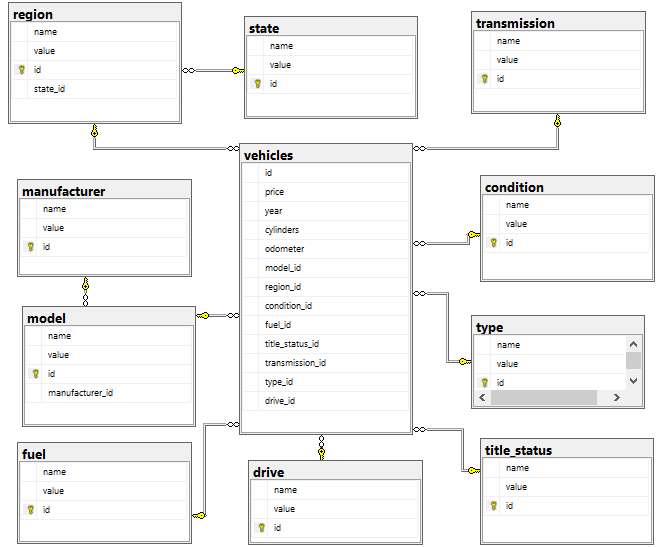


Рис. 8. Структура Бази даних Автомобілей.

Таблиця 1. Transmission

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва трансмісії | nvarchar | 50 |  |
| value | Середня вартість за трансмісією | float |  |  |

Таблиця 2. Condition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва стану | nvarchar | 50 |  |
| value | Середня вартість за станом | float |  |  |

Таблиця 3. Type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва типу | nvarchar | 50 |  |
| value | Середня вартість за типом | float |  |  |

Таблиця 4. Title\_Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва статусу | nvarchar | 50 |  |
| value | Середня вартість за статусом | float |  |  |

Таблиця 5. Drive

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва приводу | nvarchar | 50 |  |
| value | Середня вартість за приводом | float |  |  |

Таблиця 6. Fuel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва палива | nvarchar | 50 |  |
| value | Середня вартість за паливом | float |  |  |

Таблиця 7. Manufacturer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва виробника | nvarchar | 50 |  |
| value | Середня вартість за виробником | float |  |  |

Таблиця 8. State

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва штату | nvarchar | 50 |  |
| value | Середня вартість за штатом | float |  |  |

Таблиця 9. Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва штату | nvarchar | 50 |  |
| value | Середня вартість за штатом | float |  |  |
| manufacturer\_id | Унікальний номер виробника моделі | int |  | Foreign Key |

Таблиця 10. Region

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| name | Назва міста | nvarchar | 50 |  |
| value | Середня вартість за містом | float |  |  |
| state\_id | Унікальний номер штату | int |  | Foreign Key |

Таблиця 11. Vehicles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поле | Інформація | Тип | Довжина | Ключ |
| id | Унікальний номер | int |  | Primary Key |
| price | Вартість авто | int |  |  |
| year | Рік виробництва | int |  |  |
| cylinders | Кількість циліндрів двигуна | int |  |  |
| odometer | Кількість миль, що проїхало авто | int |  |  |
| model\_id | Унікальний номер моделі | int |  | Foreign Key |
| region\_id | Унікальний номер міста | int |  | Foreign Key |
| condition\_id | Унікальний номер стану | int |  | Foreign Key |
| fuel\_id | Унікальний номер палива | int |  | Foreign Key |
| title\_status\_id | Унікальний номер статусу | int |  | Foreign Key |
| transmission\_id | Унікальний номер трансмісії | int |  | Foreign Key |
| type\_id | Унікальний номер типу авто | int |  | Foreign Key |
| drive\_id | Унікальний номер приводу | int |  | Foreign Key |

Для отримання даних для аналізу скористаємося наступним запитом:  
SELECT dbo.vehicles.id, dbo.vehicles.year, dbo.vehicles.cylinders,

dbo.vehicles.odometer, dbo.vehicles.price,

dbo.condition.value AS condition,

dbo.drive.value AS drive,

dbo.fuel.value AS fuel,

dbo.manufacturer.value AS manufacturer,

dbo.model.value AS model,

dbo.region.value AS region,

dbo.state.value AS state,

dbo.title\_status.value AS title\_status,

dbo.transmission.value AS transmission,

dbo.type.value AS type

FROM dbo.vehicles INNER JOIN dbo.model ON dbo.vehicles.model\_id=dbo.model.id

INNER JOIN dbo.manufacturer ON dbo.model.manufacturer\_id= dbo.manufacturer.id

INNER JOIN dbo.region ON dbo.vehicles.region\_id = dbo.region.id

INNER JOIN dbo.state ON dbo.state.id = dbo.region.state\_id

INNER JOIN dbo.condition ON dbo.condition.id = dbo.vehicles.condition\_id

INNER JOIN dbo.drive ON dbo.drive.id = dbo.vehicles.drive\_id

INNER JOIN dbo.fuel ON dbo.fuel.id = dbo.vehicles.fuel\_id

INNER JOIN dbo.title\_status ON dbo.title\_status.id = dbo.vehicles.title\_status\_id

INNER JOIN dbo.transmission ON dbo.transmission.id = dbo.vehicles.transmission\_id

INNER JOIN dbo.type ON dbo.type.id = dbo.vehicles.type\_id

## **1.5 Опис методів аналізу даних**

Для вирішення задачі регресії Apache Spark надає нам наступні методи аналізу даних:

**Лінійна регресія.**

У статистиці лінійна регресія - це лінійний підхід до моделювання взаємозв'язку між скалярною реакцією та однією або кількома пояснювальними змінними (також відомими як залежні та незалежні змінні). Випадок однієї пояснювальної змінної називається простою лінійною регресією; більше ніж один процес називається багаторазовою лінійною регресією.

**Дерево рішень.**

Навчання на дереві рішень - один із підходів прогнозного моделювання, що використовується у статистиці, видобутку даних та машинному навчанні. Він використовує дерево рішень, щоб перейти від спостережень за предметом (представленим у гілках) до висновків про цільову вартість товару (представлених у листках). Дерева рішень, де цільова змінна може приймати безперервні значення (як правило, реальні числа), називаються деревами регресії. Дерева рішень є одними з найпопулярніших алгоритмів машинного навчання, враховуючи їх зрозумілість та простоту.

**Регресія випадкового лісу.**

Випадкові ліси або ліси випадкових рішень - це ансамблевий метод навчання для класифікації, регресії та інших завдань, які діють шляхом побудови безлічі дерев рішень під час навчання та виведення класу, що є режимом класів або середнього прогнозу окремих дерев. Випадкові ліси, як правило, перевершують дерева рішень, але їх точність нижча, ніж підсилені градієнтами дерева. Однак характеристики даних можуть впливати на їх ефективність.

**Дерево з посиленим градієнтом.**

Посилення градієнта - це техніка машинного навчання для проблем регресії та класифікації, яка створює модель прогнозування у вигляді ансамблю слабких моделей прогнозування, як правило, дерев рішень. Він будує модель поетапно, як це роблять інші методи підвищення, і узагальнює їх, дозволяючи оптимізувати довільну диференційовану функцію втрат.

**Регресія виживання.**

Аналіз виживання - це розділ статистики для аналізу очікуваної тривалості часу до того, як відбудеться одна або кілька подій, таких як смерть в біологічних організмах та несправність в механічних системах. Ця тема називається теорією надійності або аналізом надійності в техніці, аналізом тривалості чи моделюванням тривалості в економіці та аналізом історії подій у соціології.

**Ізотонічна регресія.**

У статистиці ізотонічна регресія або монотонна регресія - це техніка підгонки лінії вільної форми до послідовності спостережень, така що вмонтована лінія скрізь не зменшується (або не збільшується) і лежить якомога ближче до спостережень.

## 

## **1.6 Дослідження ефективності методів аналізу даних**

Для аналізу даних та створення програми для прогнозування ціни на автомобіль були обрані наступні методи:

* Лінійна регресія, у якості найгіршого методу.
* Регресія випадкового лісу, у якості середнього методу.
* Дерево з посиленим градієнтом, у якості найкращого методу.

**Початок дослідження**

Для дослідження методів була використана мова програмування Python та бібліотека PySpark.

Дані необхідні для аналізу збережені у course\_work.csv.

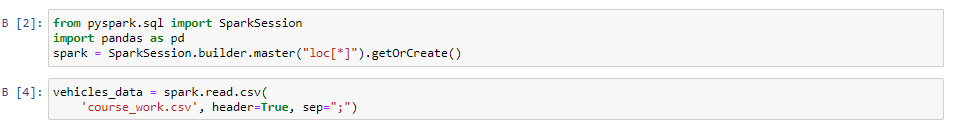


Рис. 9. Читання даних до DataFrame змінної.

**Перетворення строкових полів датасету**

Аналіз даних потребує чисельних змінних у датасеті, тому виконується перевірка на зчитані дані з файлу course\_work.csv.

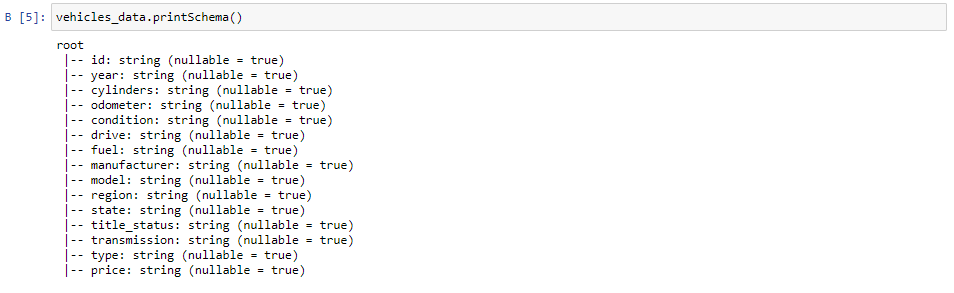


Рис. 10. Тип параметрів датасету після зчитування з файлу.

Усі параметри мають тип string, тому потрібно конвертування змінних до типів int або float.

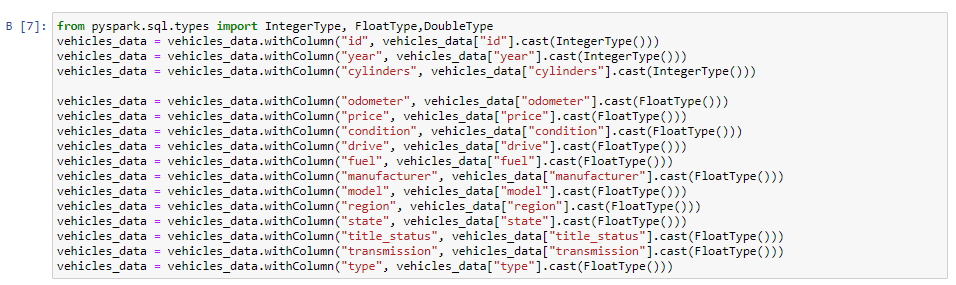


Рис. 11. Конвертування змінних датасету.

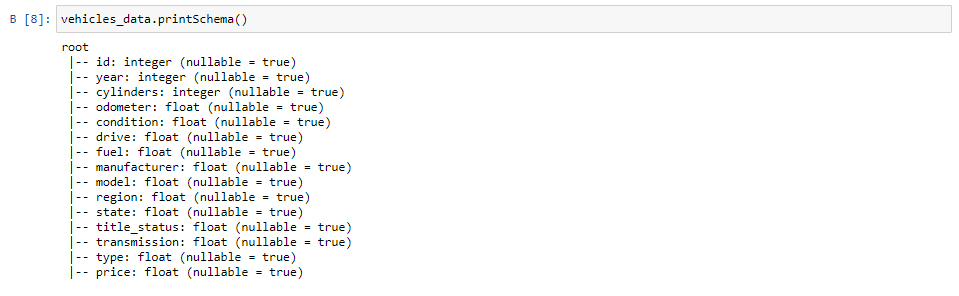


Рис. 12. Тип параметрів датасету після конвертації.

Методи аналізу даних за допомогою бібліотеки PySpark потребують перетворення даних до векторної форми.

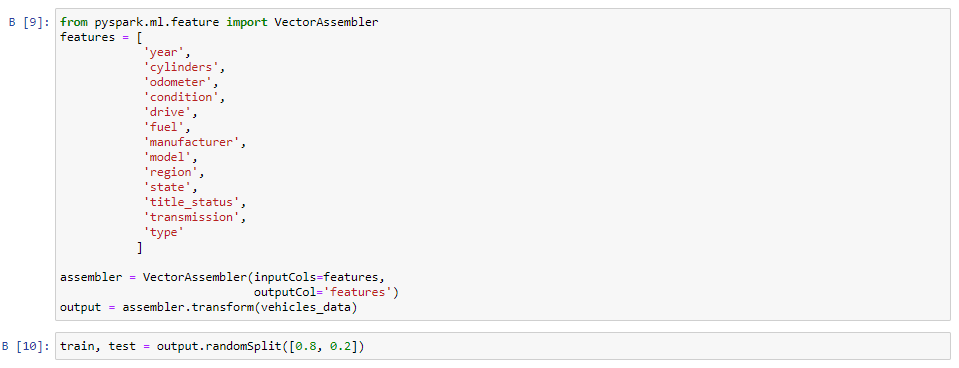


Рис. 13. Перетворення даних до векторної форми.

Для тренування моделі та тестування правильності поділяємо дані на дві групи у відношенні 80% на 20%.

**Аналіз даних за допомогою лінійної регресії**

Якщо звернути увагу на залежність між параметрами та вартістю автомобіля, то можна побачити, що лінійної залежності немає.

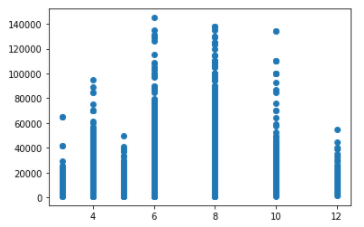


Рис. 14. Залежність ціни від кількості циліндрів у двигуні.

Тому можна зробити висновок, що метод лінійної регресії для даного датасету буде найгіршим.

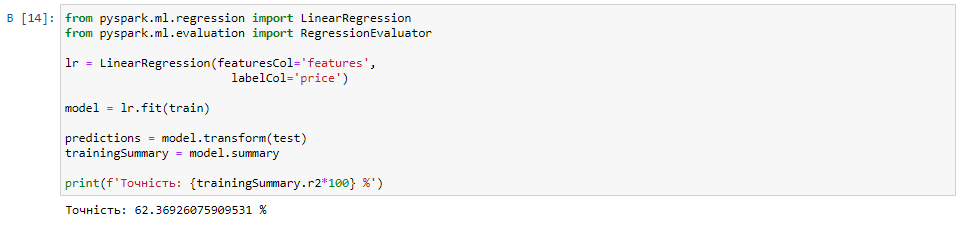


Рис. 15. Метод лінійної регресії.

Точність визначення вартості автомобіля на основі даних з датасету за допомогою методу лінійної регресії складає 63%. Для програмного забезпечення, що повинно прогнозувати вартість авто, це достатньо низька точність.

Середнєквадратична помилка методу лінійної регресії на даному датасеті дорівнює 7533. В залежності від ціни похибка може бути різною. Для прикладу візьмемо 5 автомобілей, їх ціни та спрогнозовані вартості. На Рис.18 зображена стовбчаста діаграма, яка показує різницю між реальною та спрогнозованою вартістю авто на цих п’яти прикладах. Можна зробити висновок, що різниця у окремих випадках дуже велика, та метод лінійної регресію впорався погано.



Рис. 16. Середнєквадратична помилка методу лінійної регресії.

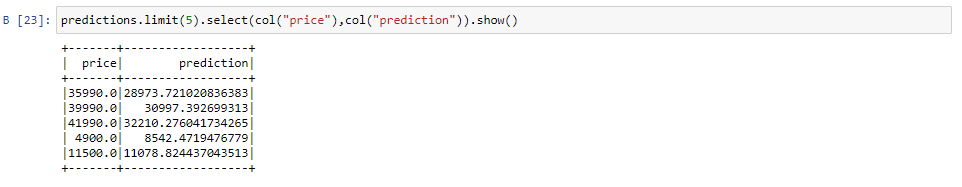


Рис. 17. Приклади спрогнозованих цін методом лінійної регресії.

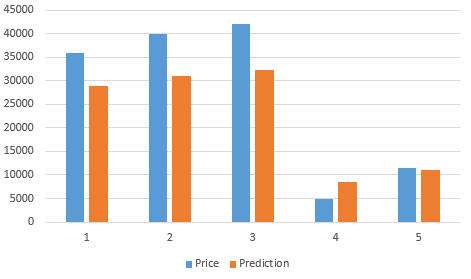


Рис. 18. Діаграма реальної та спрогнозованої вартості автомобіля за допомогою лінійної регресії.

**Аналіз даних за допомогою регресії випадкового лісу**

Алгоритм, в основі якого лежить дерево прийняття рішень, повинен краще справитися з поставленою задачею.

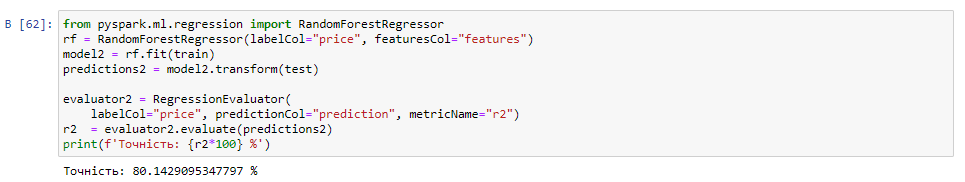


Рис. 19. Метод регресії випадкового лісу.

Точність визначення вартості автомобіля на основі даних з датасету за допомогою методу регресії випадкового лісу складає 80%. Це на 17% вище за попередній метод. Така точність набагато краще підходить для прогнозування.

Середнєквадратична помилка методу випадкового лісу на даному датасеті дорівнює 5401, що набагато менше ніж на попередньому методі. На Рис.22 зображена стовбчаста діаграма, яка показує різницю між реальною та спрогнозованою вартістю авто на цих п’яти прикладах. Можна зробити висновок, що різниця стала маленька, метод регресії випадкового лісу працює добре.

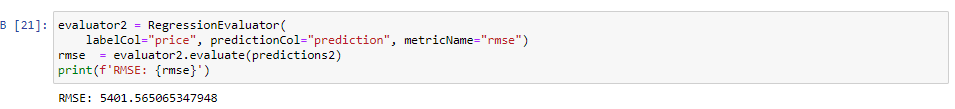


Рис. 20. Середнєквадратична помилка регресії випадкового лісу.

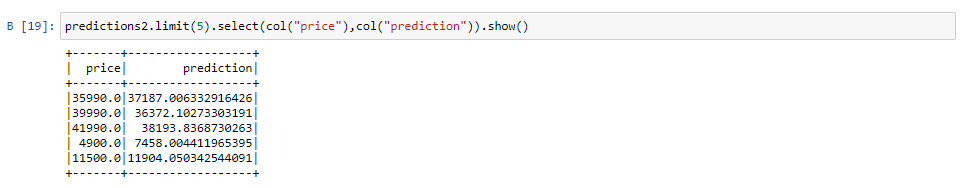


Рис. 21. Приклади спрогнозованих цін методом регресії випадкового лісу.

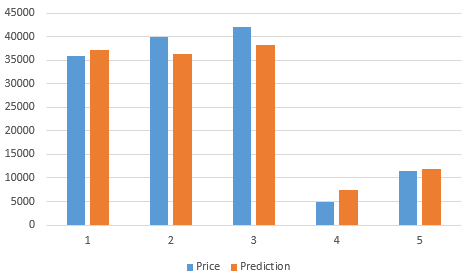


Рис. 22. Діаграма реальної та спрогнозованої вартості автомобілей методом регресії випадкового лісу.

**Аналіз даних за допомогою дерева з посиленим градієнтом**

У основі даного алгоритму також лежить дерево прийняття рішень. Точність цього алгоритму повинна бути найвищою серед інших.



Рис. 23. Метод регресії дерева з посиленим градієнтом.

Точність визначення вартості автомобіля на основі даних з датасету за допомогою методу регресії дерева з посиленим градієнтом складає 85%. Це на 5% вище за попередній метод. Така точність є найкращою серед усіх досліджених методів.

Середнєквадратична помилка методу випадкового лісу на даному датасеті дорівнює 4773, що є найкращим результатом серед усіх методів. На Рис.22 зображена стовбчаста діаграма, яка показує різницю між реальною та спрогнозованою вартістю авто на цих п’яти прикладах.

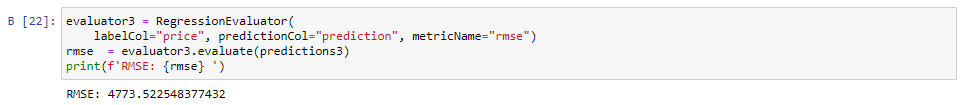


Рис. 24. Середнєквадратична помилка регресії дерева з посиленим градієнтом.

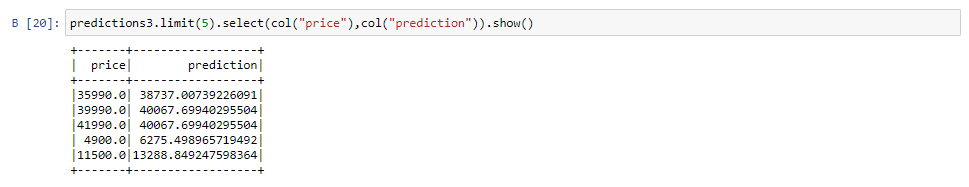


Рис. 25. Приклади спрогнозованих цін методом регресії дерева з посиленим градієнтом.

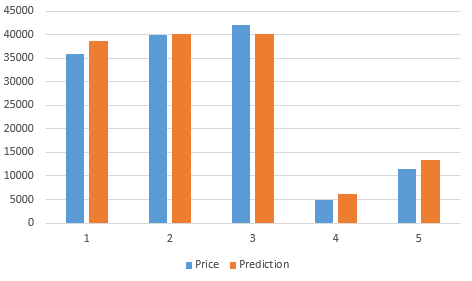


Рис. 26. Діаграма реальної та спрогнозованої вартості автомобілей методом регресії дерева з посиленим градієнтом.

У результаті дослідження трьох методів регресії для прогнозування вартості автомобіля, у якості найкращого метода можна виділити регресію дерева з посиленим градієнтом, точність якої дорівнює 85%.

# **ВИСНОВОК**

При виконанні курсової роботи була вирішена задача прогнозування цін на автомобілі. Були виконані: пошук датасету з параметрами машин, попередня обробка датасету для усунення неважливих параметрів та пустих рядків, розробка БД для зручного зберігання даних, опис методів аналізу та дослідження методів аналізу. Для дослідження були обрані методи лінійної регресії, випадкового лісу та дерева з посиленим градієнтом. Результатом досліджень є дані у таблиці 12.

Таблиця 12. Результати дослідження методів.

|  |  |  |
| --- | --- | --- |
| Назва методу | Точність | RMSE |
| Лінійна регресія | 62% | 7533 |
| Регресія випадкового лісу. | 80.8% | 5401 |
| Регресія дерева з посиленим градієнтом. | 85% | 4773 |

Метод лінійної регресії виявився найгіршим, результати тренування моделі неможливо використовувати у прогнозуванні вартості автомобілей. Регресія випадкового лісу опрацювала дані дуже добре, результати тренування моделі можна використовувати у прогнозуванні, проте буде похибка. Дослідження регресія дерева з посиленим градієнтом показало, що цей метод є найкращим, результати тренування моделі можна використовувати у прогнозуванні, похибка також буде, проте значно менше.

# **ПЕРЕЛІК ПОСИЛАНЬ**

1. Аналіз малих даних. Python: категоріальні признаки. [Електронний ресурс]  
   Доступ до ресурсу: https://dyakonov.org/2016/08/03/python-категориальные-признаки/
2. David A. Freedman. Statistical Models: Theory and Practice. 2009, 26 c.
3. Wu, Xindong; Kumar, Vipin; Ross Quinlan, J.; Ghosh, Joydeep; Yang, Qiang; Motoda, Hiroshi; McLachlan, Geoffrey J.; Ng, Angus; Liu, Bing; Yu, Philip S.; Zhou, Zhi-Hua. "Top 10 algorithms in data mining". 2008
4. Ho, Tin Kaam. Random decision Forests. 1995, 5 c.
5. Breiman L. Arcing the Edge. 1997, 14 c.
6. Towards data science. Isotonic regression. [Електронний ресурс]  
   Доступ до ресурсу: https://towardsdatascience.com/isotonic-regression-is-the-coolest-machine-learning-model-you-might-not-have-heard-of-3ce14afc6d1e
7. NCBI. Survival and regression models. [Електронний ресурс]  
   Доступ до ресурсу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4111957/

# **ДОДАТОК А ТЕКСТИ ПРОГРАМНОГО ПРОДУКТУ**

from pyspark.sql import SparkSession

from pyspark.sql.functions import col

import pandas as pd

spark = SparkSession.builder.master("loc[\*]").getOrCreate()

vehicles\_data = spark.read.csv(

'course\_workF.csv', header=True, sep=";")

vehicles\_data.printSchema()

from pyspark.sql.types import IntegerType, FloatType,DoubleType

vehicles\_data = vehicles\_data.withColumn("id", vehicles\_data["id"].cast(IntegerType()))

vehicles\_data = vehicles\_data.withColumn("year", vehicles\_data["year"].cast(IntegerType()))

vehicles\_data = vehicles\_data.withColumn("cylinders", vehicles\_data["cylinders"].cast(IntegerType()))

vehicles\_data = vehicles\_data.withColumn("odometer", vehicles\_data["odometer"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("price", vehicles\_data["price"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("condition", vehicles\_data["condition"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("drive", vehicles\_data["drive"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("fuel", vehicles\_data["fuel"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("manufacturer", vehicles\_data["manufacturer"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("model", vehicles\_data["model"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("region", vehicles\_data["region"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("state", vehicles\_data["state"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("title\_status", vehicles\_data["title\_status"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("transmission", vehicles\_data["transmission"].cast(FloatType()))

vehicles\_data = vehicles\_data.withColumn("type", vehicles\_data["type"].cast(FloatType()))

from pyspark.ml.feature import VectorAssembler

features = [

'year',

'cylinders',

'odometer',

'condition',

'drive',

'fuel',

'manufacturer',

'model',

'region',

'state',

'title\_status',

'transmission',

'type'

]

assembler = VectorAssembler(inputCols=features,

outputCol='features')

output = assembler.transform(vehicles\_data)

train, test = output.randomSplit([0.8, 0.2])

from pyspark.ml.regression import LinearRegression

from pyspark.ml.evaluation import RegressionEvaluator

lr = LinearRegression(featuresCol='features',

labelCol='price')

model = lr.fit(train)

predictions = model.transform(test)

trainingSummary = model.summary

print(f'Точність: {trainingSummary.r2\*100} %')

print("RMSE: %f" % trainingSummary.rootMeanSquaredError)

predictions.limit(5).select(col("price"),col("prediction")).show()

from pyspark.ml.regression import RandomForestRegressor

rf = RandomForestRegressor(labelCol="price", featuresCol="features")

model2 = rf.fit(train)

predictions2 = model2.transform(test)

evaluator2 = RegressionEvaluator(

labelCol="price", predictionCol="prediction", metricName="r2")

r2 = evaluator2.evaluate(predictions2)

print(f'Точність: {r2\*100} %')

evaluator2 = RegressionEvaluator(

labelCol="price", predictionCol="prediction", metricName="rmse")

rmse = evaluator2.evaluate(predictions2)

print(f'RMSE: {rmse}')

predictions2.limit(5).select(col("price"),col("prediction")).show()

from pyspark.ml.regression import GBTRegressor

gf = GBTRegressor(labelCol="price", featuresCol="features")

model3 = gf.fit(train)

predictions3 = model3.transform(test)

evaluator3 = RegressionEvaluator(

labelCol="price", predictionCol="prediction", metricName="r2")

r2 = evaluator3.evaluate(predictions3)

print(f'Точность: {r2\*100} %')

evaluator3 = RegressionEvaluator(

labelCol="price", predictionCol="prediction", metricName="rmse")

rmse = evaluator3.evaluate(predictions3)

print(f'RMSE: {rmse} ')

predictions3.limit(5).select(col("price"),col("prediction")).show()