DOI: https://doi.org/10.36910/6775-2524-0560-2023-53-30
УДК: 004.58
Поліщук Микола Миколайович, к.т.н., доцент
https://orcid.org/0000-0002-1218-5925
Цибень Дмитро Васильович, магістрант
Каплюк Юрій Іванович, магістрант
Луцький національний технічний університет, м. Луцьк, Україна

ОБРОБКА ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ МАШИНОГО НАВЧАННЯ ЗАСОБАМИ PYTHON

Поліщук М.М., Цибень Д.В., Карплюк Ю.І. Обробка інформації за допомогою машиного навчання засобами Python. У статті докладно розглядаються основні кроки та процеси використання машинного навчання для створення рекомендаційних систем. Розглянуто, як зберегти дані, навчити модель та створити зручний інструмент для рекомендацій жанрів фільмів на основі індивідуальних вподобань користувачів. Продемонстровано ключові кроки, від відбору даних до передбачення рекомендацій, як використовувати Python та потужні бібліотеки для досягнення цієї мети.
Ключові слова: машинне навчання, Python, обробка даних, персоналізовані рекомендації.

Polishchuk M., Tsyben D., Karpluk Y. Information processing using machine learning using Python. The article discusses in detail the main steps and processes of using machine learning to create recommender systems. It looks at how to store data, train a model, and create a user-friendly tool for recommending movie genres based on individual user preferences. The key steps, from data selection to recommendation prediction, are demonstrated on how to use Python and powerful libraries to achieve this goal.
Key words: machine learning, Python, data processing, personalized recommendations.

Вступ. У світі, де величезні обсяги даних надходять щоденно з найрізноманітніших джерел, ефективна обробка цієї інформації та виділення корисних з них стає важливим завданням [1]. Завдяки розвитку машинного навчання, ми маємо унікальну можливість використовувати алгоритми та моделі для автоматичного аналізу цих даних. Проте, існує низка проблем, які стоять перед дослідниками та фахівцями у цій області. Перша з них полягає у виборі найбільш підходящих методів та алгоритмів машинного навчання для конкретних завдань обробки даних. Друга проблема полягає у впровадженні та оптимізації цих методів з використанням Python, який став однією з найпопулярніших мов програмування для цієї цілі. У цьому контексті, поява питань про те, як ефективно використовувати інструменти машинного навчання на основі Python для обробки інформації та вирішення реальних завдань стає досить актуальною.

Постановка проблеми. Для більш детального розуміння постановки проблеми, слід відзначити, що в сучасному світі дані великої обсягу стають надзвичайно цінним ресурсом. Навряд чи знайдеться галузь, де б не було потреби в обробці та аналізі даних, і ця потреба постійно зростає. Водночас, Python, завдяки своїй легкості використання та багатству бібліотек для машинного навчання, став популярним інструментом для розв'язання цих завдань.

Проте, існують питання про оптимальність вибору методів та підходів для обробки даних в Python, а також про практичне застосування машинного навчання в реальних задачах. Отже, в цьому контексті, наша проблема полягає в розробці підходів і стратегій для оптимізації використання Python та інструментів машинного навчання для обробки інформації, а також у визначенні найкращих практик і методів вирішення проблем, пов'язаних з цією областю.

Викладення основного матеріалу. Машинне навчання (ML) представляє собою галузь штучного інтелекту, яка зосереджена на розвитку алгоритмів і моделей, які дозволяють системам автоматично набувати знання і навички з великих обсягів даних. Головна ідея полягає в тому, щоб система могла "навчатися" на даних та робити передбачення або приймати рішення без явної програмної інструкції.

Машинне навчання відіграє ключову роль у сфері обробки інформації [2], оскільки дозволяє автоматизувати процеси аналізу та використовувати великі обсяги даних для виявлення патернів, класифікації і прогнозування. Це застосовується в багатьох сферах, включаючи медицину, фінанси, маркетинг, транспорт, технології та багато інших. Python, завдяки своїм багатим бібліотекам і простоті використання, став основним інструментом для розробки і впровадження моделей машинного навчання у сфері обробки інформації, що дозволяє фахівцям здійснювати аналіз та отримувати цінні висновки з даних швидше та ефективніше [3].

Python відіграє визначальну роль у сфері машинного навчання з кількох важливих причин:

- Python відіграє визначальну роль у сфері машинного навчання з кількох важливих причин:
- Простота вивчення та використання: Python відомий своєю простотою та легкістю вивчення для початківців. Ця мова програмування має зрозумілий синтаксис та багатий набір бібліотек, що робить її доступною для широкого кола користувачів.
- Багатий екосистема бібліотек: Python має широкий вибір бібліотек, спеціально призначених для розробки машинного навчання та обробки даних. Бібліотеки, такі як NumPy, Pandas, scikit-learn, TensorFlow і PyTorch, надають інструменти для вивчення, аналізу та створення моделей машинного навчання.
- Загальна популярність та підтримка спільноти: Python є однією з найпопулярніших мов програмування в світі, що означає наявність великої активної спільноти розробників. Ця спільнота надає безліч ресурсів, від відкритих курсів до форумів підтримки, що сприяє обміну знаннями та розв'язанню проблем.
- Портативність та сумісність: Python працює на різних операційних системах і може легко інтегруватися з іншими мовами, що сприяє розробці масштабованих та портативних застосунків.
- Індустріальні застосування: Python широко використовується в промислових рішеннях для машинного навчання та обробки даних. Це означає, що компанії і організації, які вкладають ресурси в розробку Python-програм, можуть легко забезпечити собі підтримку та довгостроковий розвиток своїх проектів.

Усі ці фактори роблять Python сильним інструментом для машинного навчання і обробки інформації, забезпечуючи швидкий старт та можливість розвивати проекти в цій галузі з високою продуктивністю.

Машинне навчання – це галузь штучного інтелекту, яка займається розробкою алгоритмів та моделей, які навчаються на основі даних та дозволяють системам виконувати завдання без явного програмного коду [4]. Основні поняття в машинному навчанні можна поділити на кілька ключових категорій:

- Навчання з учителем (Supervised Learning). Це тип машинного навчання, де алгоритм навчається на основі пари вхідних даних і відповідних вихідних даних (ярликів). Іншими словами, алгоритм "навчається" на основі прикладів, де маємо вхідні дані та відомі правильні відповіді. Навчання з учителем використовується для задач, таких як класифікація (призначення об'єкта до однієї з кількох категорій) та регресія (прогнозування числового значення).
- Навчання без учителя (Unsupervised Learning). У цьому типі машинного навчання модель навчається на даних без явно визначених відповідей. Алгоритми намагаються знайти приховані структури або патерни в наборі даних. Навчання без учителя використовується для задач, таких як кластеризація (групування подібних об'єктів) та зменшення розмірності (скорочення кількості ознак, зберігаючи корисну інформацію).
- Підготовка даних (Data Preprocessing). Це процес очищення, перетворення та підготовки даних перед тим, як вони використовуються для навчання моделі. Включає в себе завдання, такі як обрізання відсутніх даних, нормалізація ознак, кодування категоріальних змінних та розділення даних на навчальний і тестовий набори. Правильна підготовка даних є критично важливою для досягнення високої якості моделі машинного навчання. Незадовільна підготовка може призвести до невірних результатів та погіршення роботи алгоритму.

Ці основні поняття є фундаментом машинного навчання та обробки даних, і їх розуміння є важливим для успішної роботи з цими технологіями.

Приклад використання штучного інтелекту для вирішення задач.

Якщо потрібно вирішити задачу: визначити ідеальний жанр фільму для конкретної людини, то для досягнення цієї мети, використовується бібліотека Scikit-learn, яка надає широкий спектр інструментів для машинного навчання та аналізу даних [5].

Спочатку необхідно зібрати та підготувати дані, які будуть використовуватися штучним інтелектом для прийняття рішень щодо рекомендацій. Це включає в себе збір інформації про користувачів, їх улюблені жанри, вік, гендер та наявність вищої освіти. Ці дані вимагають обробки та перетворення в придатний для подальшого аналізу формат.

Для успішного вирішення завдання, першим та важливим етапом є генерація файлу у форматі CSV за допомогою бібліотеки Pandas. Цей файл буде важливим джерелом даних, які необхідні для подальшого аналізу та обробки. Пакет Pandas, завдяки своїм потужним інструментам для роботи з даними, дозволить мені створити структурований набір інформації, який буде включати дані про вік, гендер та наявність вищої освіти.

```
import pandas as pd # імпортуєм пакет pandas для генерації CSV файлу
# вставляєм в функцію DataFrame підготовлені завчасно дані
df = pd.DataFrame({'Age': [20, 25, 25, 35, 20, 25, 25, 35]
                   'Sex': [0, 0, 0, 0, 1, 1, 1, 1], # 0 - Male, 1 – Female
                   'Higher Education': [False, True, False, False, False, True, False, False],
                   'Genre': ['Action', 'Historical', 'Comedy', 'Comedy', 'Romance', 'Drama', 'Romance',
'Drama']})
# записуєм дані в csv
df.to_csv('prepared_data.csv')
```

```
>>> df
   Age  Sex  Higher Education      Genre
0   20    0             False     Action
1   25    0              True  Historical
2   25    0             False     Comedy
3   35    0             False     Comedy
4   20    1             False    Romance
5   25    1              True      Drama
6   25    1             False    Romance
7   35    1             False      Drama
```

Рис. 1. Вивід CSV в терміналі

Після успішної генерації та підготовки даних у форматі CSV, настав час перейти до наступного важливого етапу – запису цих даних в модель за допомогою бібліотеки Scikit-learn (sklearn). Scikit-learn - це потужна бібліотека для машинного навчання та статистичного аналізу даних, яка надає широкий спектр інструментів для створення, навчання та оцінки моделей.

```
from sklearn.tree import DecisionTreeClassifier # Імпорт класу із пакета sklearn
data = pd.read_csv('prepared_data.csv') # читання CSV файла
X = data.drop(columns=['Genre'])
y = data['Genre']
model = DecisionTreeClassifier() # виклик класу
model.fit(X.values, y)
```

Після успішного навчання моделі за допомогою бібліотеки Scikit-learn, настав час перейти до завершального кроку – збереження цієї моделі за допомогою бібліотеки joblib. joblib – це потужний інструмент для збереження та завантаження об'єктів Python, включаючи навчені моделі машинного навчання.

Зберігання моделі за допомогою joblib дозволить мені зберегти результати навчання та параметри моделі для подальшого використання без необхідності повторного навчання. Це особливо корисно, якщо моя рекомендаційна система має працювати в реальному часі та постійно оновлюватися.

```
import joblib
joblib.dump(model, 'film-recommender.joblib') # збереження моделі
```

Тепер є збереженна модель, завдяки якій можна перейти до виконання прогнозів. Я використаю навчену модель та передам в неї вхідну інформацію про конкретного користувача через функцію input(). Модель аналізуватиме цю інформацію і рекомендуватиме найкращий жанр фільмів для даного користувача.

Предікт та рекомендації стануть результатом використання машинного навчання та аналізу даних для створення персоналізованих рекомендаційних систем, які допоможуть користувачам знайти найкращий жанр фільмів, який найкраще відповідають їхнім індивідуальним смакам та вподобанням (рис. 2).

```
def predict_genre_for_user():
    model = joblib.load("film-recommender.joblib")
    age = input("Your age: ")
    sex = input("Your gender (0 - male, 1 - female): ")
    education = input("Do you have a higher education? (True/False): ")
    pred = model.predict([[age, sex, bool(education)]])[0]
    return pred

predict_genre_for_user()
```

```
>>> predict_genre_for_user()
Your age: 25
Your gender (0 - male, 1 - female): 0
Do you have a higher education? (True/False): True
'Historical'
>>> predict_genre_for_user()
Your age: 22
Your gender (0 - male, 1 - female): 0
Do you have a higher education? (True/False): False
'Action'
>>>
```

Рис. 2. Вивід роботи функції predict_genre_for_user()

Також можна продемонструвати дерево рішень (рис. 3) по якому орієнтує ця штучний інтелект при виборі жанру фільмів для користувача, використовуючи даний код:

```
from sklearn import tree
tree.export_graphviz(model, out_file="film-recommender.dot",
                     feature_names=["Age", "Sex", "Higher Education"],
                     class_names=sorted(y.unique()),
                     label="all",
                     rounded=True,
                     filled=True)
```

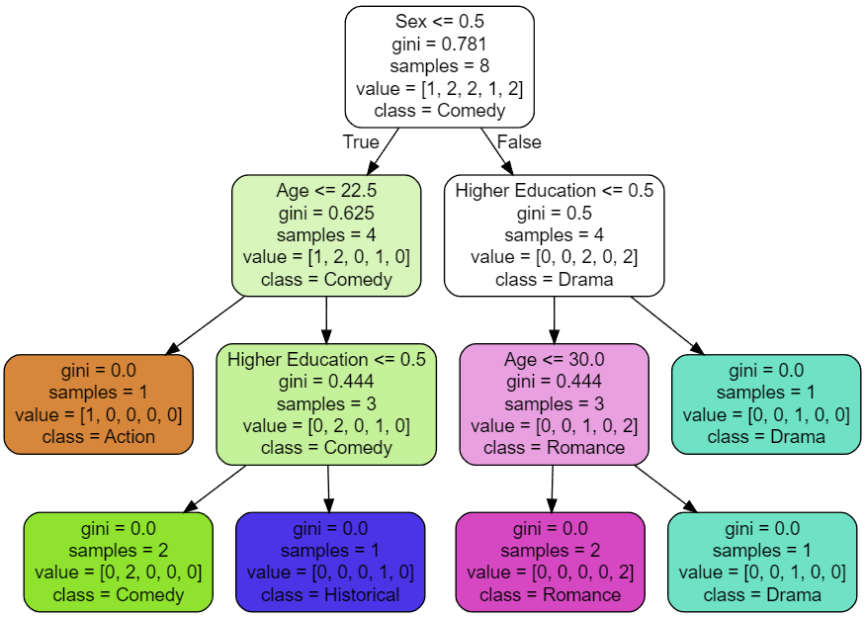

Рис. 3. Дерево рішень

Висновки.

У ході виконання роботи було розглянуто використання машинного навчання та мови програмування Python для обробки інформації. Ми розпочали з постановки завдання, підготовки даних та навчання моделі. Ми вивчили процес обробки та аналізу даних, включаючи підготовку та навчання моделі, а також збереження її для подальшого використання. Рекомендаційні системи, створені з використанням машинного навчання та Python, відкривають безмежні можливості для надання користувачам персоналізованих рекомендацій та покращення їхнього досвіду в споживанні контенту.

Список бібліографічного опису

1. Коротка історія штучного інтелекту. Український тиждень. URL: https://tyzhden.ua/korotka-istoriia-shtuchnoho-intelektu/ (дата звернення: 5.09.2023).
2. The history of artificial intelligence - science in the news. Science in the News. URL: https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/ (дата звернення: 6.09.2023)
3. Machine Learning, ML. IT-Enterprise – your one-stop ecosystem for reengineering | it.ua. URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning (дата звернення: 10.09.2023).
4. Pedamkar P. Machine Learning Methods | Types of Classification in Machine Learning. EDUCBA. URL: https://www.educba.com/machine-learning-methods/ (дата звернення: 20.09.2023).
5. Machine learning: the problem setting. scikit-learn. URL: https://scikit-learn.org/0.21/tutorial/basic/tutorial.html (дата звернення: 10.10.2023).

References

1. A brief history of artificial intelligence. Ukrainian week. URL: https://tyzhden.ua/korotka-istoriia-shtuchnoho-intelektu/ (access date: 09/5/2023).
2. The history of artificial intelligence - science in the news. Science in the News. URL: https://sitn.hms.harvard.edu/flash/2017/history-artificial-intelligence/ (access date: 09/6/2023).
3. Machine Learning, ML. IT-Enterprise – your one-stop ecosystem for reengineering | it.ua URL: https://www.it.ua/knowledge-base/technology-innovation/machine-learning (access date: September 10, 2023).
4. Pedamkar P. Machine Learning Methods Types of Classification in Machine Learning. EDUCBA. URL: https://www.educba.com/machine-learning-methods/ (date accessed: 09/20/2023).
5. Machine learning: the problem setting. scikit-learn. URL: https://scikit-learn.org/0.21/tutorial/basic/tutorial.html (accessed 10/10/2023).