

ІНТЕГРАЦІЯ ШТУЧНОГО ІНТЕЛЕКТУ У VISUAL STUDIO ЗА ДОПОМОГОЮ ML.NET

Федченко Л. Ю. – гр. БІП-1-24, студентка, fedchenkoludya@gmail.com

Пилипенко В. І. – старший викладач, pylypenko.vi@knu.edu.ua

Київський національний університет технологій та дизайну

Метою роботи є дослідження можливостей ML.NET для реалізації моделей машинного навчання у середовищі C#.

ML.NET – це фреймворк від Microsoft, який дозволяє розробникам C# та .NET створювати, тестувати та впроваджувати моделі машинного навчання без необхідності переходити на інші мови [1]. Фреймворк інтегрується з Visual Studio та підтримує широкий спектр ML-задач: класифікацію, регресію, кластеризацію, NLP та аналіз зображень. Крім того, ML.NET сумісний з ONNX, що дає можливість імпортувати моделі з TensorFlow або PyTorch [2]. На рис.1 представлено вікно середовища розробки із сценаріями у ML.NET.

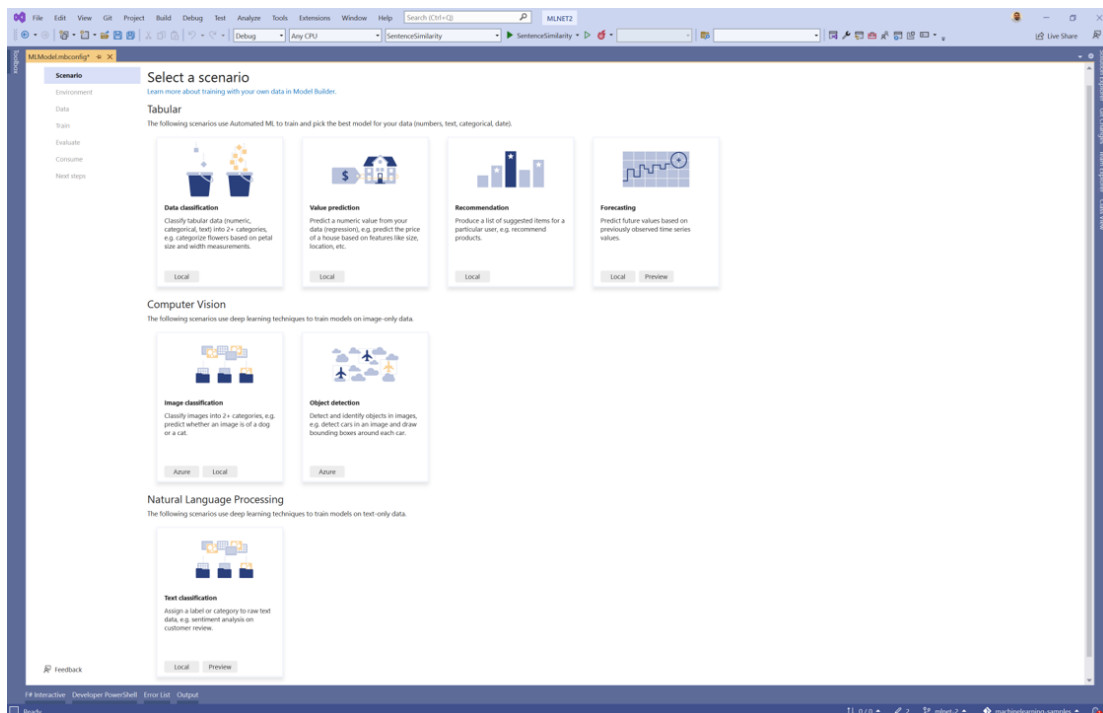


Рисунок 1 – Сценарії ML.NET у Model Builder

Однією з основних переваг ML.NET є його підтримка широкого спектра задач машинного навчання, що дозволяє розробникам реалізувати різноманітні сценарії без необхідності використання сторонніх мов програмування. Можна

Платформа: ІНФОРМАЦІЙНІ СИСТЕМИ. КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ. ТЕХНОЛОГІЇ INTERNET OF THINGS TA SMART-СИСТЕМИ

виокремити декілька типів задач, які підтримує ML.NET: класифікація, регресія, кластеризація, аналіз зображень [3].

На рис. 2 представлено програмний код по інтеграції ML.NET для класифікації тональності тексту. А саме передбачає, чи є відгук позитивним чи негативним.

```
// Створення ML-контексту
var context = new MLContext();

// Завантаження даних
string dataPath = "sentiment_data.tsv";
var data = context.Data.LoadFromTextFile<SentimentData>(
    dataPath, hasHeader: true, separatorChar: '\t');

// Побудова пайплайну обробки
var pipeline = context.Transforms.Text.FeaturizeText(
    outputColumnName: "Features", inputColumnName: nameof(SentimentData.Text))
    .Append(context.BinaryClassification.Trainers.SdcaLogisticRegression(
        labelColumnName: "Label", featureColumnName: "Features"));

// Тренування моделі
var model = pipeline.Fit(data);

// Створення предиктора
var predictionEngine = context.Model.CreatePredictionEngine<SentimentData,
    SentimentPrediction>(model);

// Тестове передбачення
var input = new SentimentData { Text = "I really love this!" };
var prediction = predictionEngine.Predict(input);
```

Рисунок 2 – Програмний код по класифікації тональності тексту

Висновок. ML.NET – це зручний і потужний інструмент для розробників .NET, який дозволяє створювати та впроваджувати моделі машинного навчання без знання сторонніх мов. Завдяки широкому функціоналу, інтеграції з Visual Studio та підтримці AutoML, ML.NET є ефективним рішенням для впровадження штучного інтелекту в C#-проекти.

Л і т е р а т у р а

1. Price, M. J. (2019). C# 8.0 and .NET Core 3.0–Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code. Packt Publishing Ltd.
2. Vasilev, I., Slater, D., Spacagna, G., Roelants, P., & Zocca, V. (2019). Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow. Packt Publishing Ltd.
3. Ahmed, Z., Amizadeh, S., Bilenko, M., Carr, R., Chin, W. S., Dekel, Y., & Zhu, Y. (2019, July). Machine learning at Microsoft with ML.NET. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2448-2458).